CYBORTS

Final Report

PYTHON-BASED INTRUSION DETECTION SYSTEM (IDS) WITH REAL-TIME MONITORING & AUTOMATED REPORTING

NAME: SHARIQ FAROOQ

DOMAIN: CYBERSECURITY & DIGITAL FORENSICS

INTERNSHIP ORGANIZATION: CYBORTS

GITHUB: HTTPS://GITHUB.COM/SHARIQFAROOQDEV

LINKEDIN: WWW.LINKEDIN.COM/IN/SHARIQ-FAROOQ

Table of Contents

Executive Summary

This report documents the development of a Python-based Intrusion Detection System (IDS) with real-time monitoring and automated reporting capabilities. Implemented during a cybersecurity internship at Cyborts, the project was executed in four phases, each building on the previous to create a fully functional IDS solution. The final output is a standalone, GUI-based system that detects, logs, and reports suspicious network activity, offering an accessible view of real-time threats for security analysts.

Project Overview

The primary goal of the project was to develop an Intrusion Detection System capable of:

- Capturing live network packets
- Matching them against custom detection rules
- Alerting users in real-time
- Logging suspicious activity
- Providing a GUI dashboard for monitoring
- Generating HTML reports with PDF export
- Archiving logs for long-term storage

The IDS was implemented in Python using libraries like scapy, tkinter, matplotlib, and jinja2. The project was modular, extensible, and suitable for small-to-medium scale SOC environments or educational use.

Module Summaries

Module 1: Packet Sniffer

- **Objective**: Capture live network traffic using Scapy.
- **Tools Used**: scapy, threading

Features:

- Interface selection
- Real-time packet capture
- Lightweight parsing (protocol, source/destination, payload)

Module 2: Rule Matching Engine

- **Objective**: Match live packets against custom flat-file rules.
- **Tools Used**: Regex, configparser

Features:

- Flat-file rule parsing (TCP/UDP/IP matches)
- Severity scoring system
- Alert generation with color-coding

Module 3: Logging & Alert System

- **Objective**: Log alerts, play audio warnings, and structure data.
- **Tools Used**: pygame, JSON, file handling

Features:

- Alerts stored in suspicious_packets.json
- Sound notifications for critical alerts
- Organized alert format (timestamp, protocol, severity, message)

Module 4: GUI Dashboard & Reporting

- **Objective**: Build a live-updating dashboard for real-time interaction.
- **Tools Used**: tkinter, matplotlib, jinja2, pdfkit

Features:

- GUI with alert table, live chart, and control buttons
- Real-time stats for TCP/UDP/ICMP/Other protocols
- HTML report generation using Jinja2 templating
- PDF export via pdfkit (wkhtmltopdf)
- Archiving system for historical logs

Final Output Features

Feature	Status
Packet Sniffing	✅ Complete
Custom Rule Engine	✅ Complete
Real-Time Alerts & Logging	✅ Complete
Audio Notifications	✅ Complete
Live GUI Dashboard	✅ Complete
Protocol Distribution Chart	✅ Complete
HTML + PDF Reporting	✅ Complete
Log Archiving System	✅ Complete
Modular & Extensible Code	✅ Complete

Conclusion

This project offered an end-to-end experience of building an intrusion detection system from scratch. By integrating real-time data capture, intelligent rule matching, user interface design, and automated reporting, the project simulates real-world SOC tools at a fundamental level. The final solution is user-friendly, informative, and modular enough for future upgrades.

Limitations

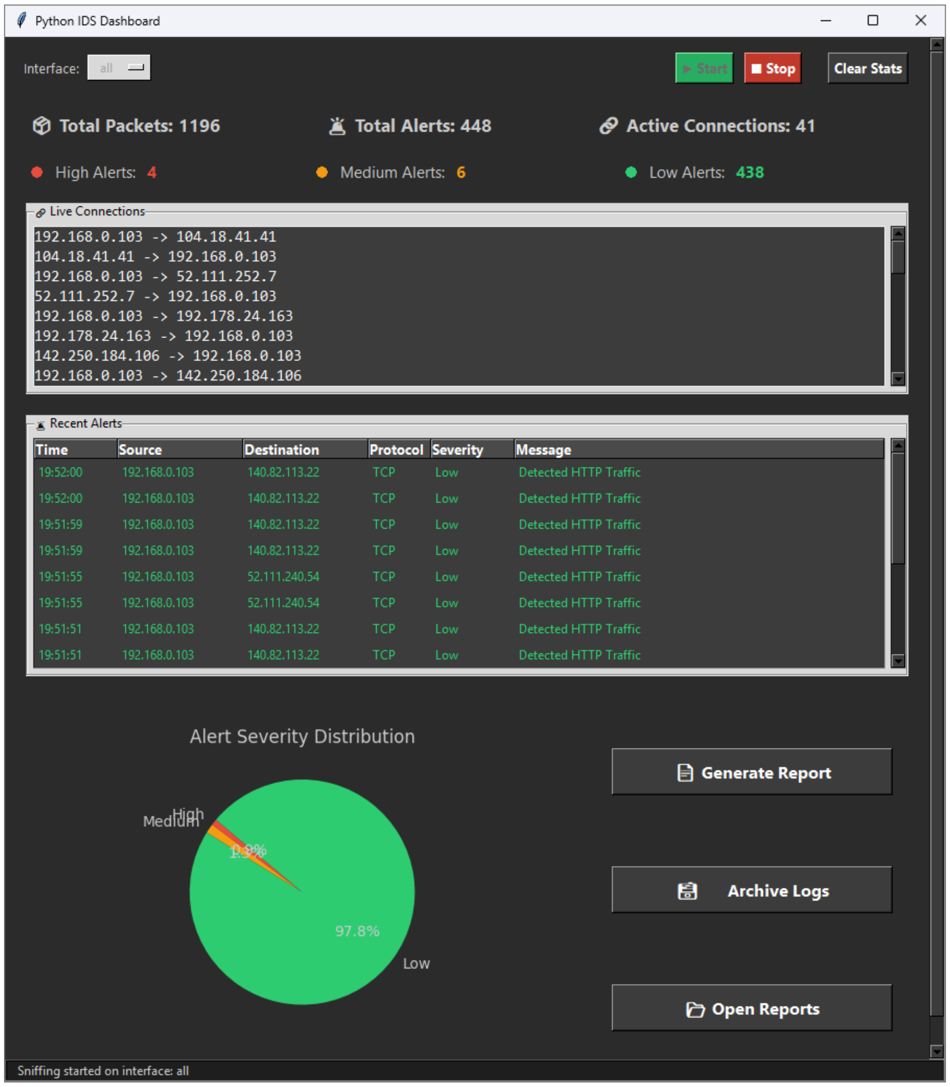
- Not compatible with encrypted traffic (e.g., HTTPS payloads)
- Basic rule engine (no ML or anomaly detection)
- GUI performance may degrade under high network load
- PDF export requires external wkhtmltopdf binary

Future Improvements

- Implement anomaly-based detection using ML models
- Support multi-user role-based dashboards
- Include GeoIP mapping for attacker IPs
- Enhance rule engine with UI-based rule editor
- Add email/SMS alert integration for SOC operators

Screenshots

a. GUI Dashboard Overview



Total Packets: 1196 | Total Alerts: 448 | Active Connections: 41

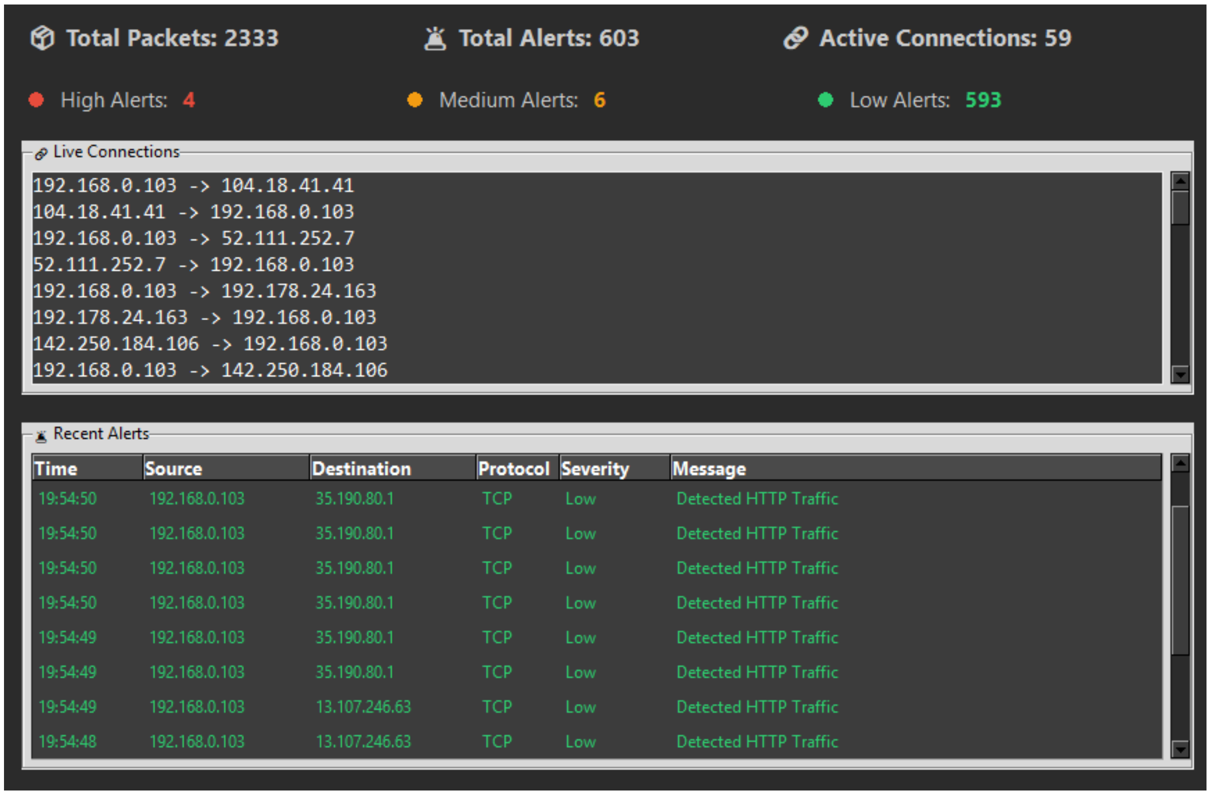
High Alerts: 4 | Medium Alerts: 6 | Low Alerts: 438

```
192.168.0.103 -> 104.18.41.41
104.18.41.41 -> 192.168.0.103
192.168.0.103 -> 52.111.252.7
52.111.252.7 -> 192.168.0.103
192.168.0.103 -> 192.178.24.163
192.178.24.163 -> 192.168.0.103
142.250.184.106 -> 192.168.0.103
192.168.0.103 -> 142.250.184.106
```

Time	Source	Destination	Protocol	Severity	Message
19:52:00	192.168.0.103	140.82.113.22	TCP	Low	Detected HTTP Traffic
19:52:00	192.168.0.103	140.82.113.22	TCP	Low	Detected HTTP Traffic
19:51:59	192.168.0.103	140.82.113.22	TCP	Low	Detected HTTP Traffic
19:51:59	192.168.0.103	140.82.113.22	TCP	Low	Detected HTTP Traffic
19:51:55	192.168.0.103	52.111.240.54	TCP	Low	Detected HTTP Traffic
19:51:55	192.168.0.103	52.111.240.54	TCP	Low	Detected HTTP Traffic
19:51:51	192.168.0.103	140.82.113.22	TCP	Low	Detected HTTP Traffic
19:51:51	192.168.0.103	140.82.113.22	TCP	Low	Detected HTTP Traffic

Sniffing started on interface: all

b. Real-time Alerts



Total Packets: 2333 Total Alerts: 603 Active Connections: 59

● High Alerts: 4 ● Medium Alerts: 6 ● Low Alerts: 593

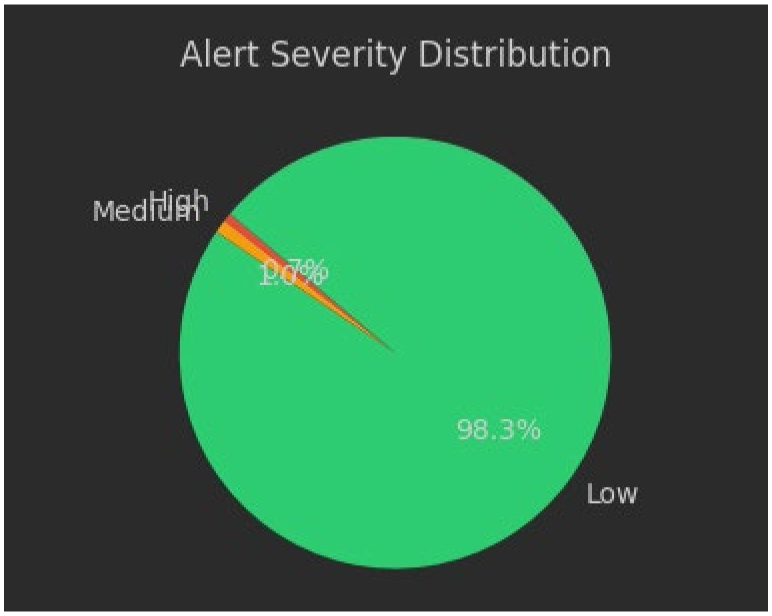
Live Connections

```
192.168.0.103 -> 104.18.41.41
104.18.41.41 -> 192.168.0.103
192.168.0.103 -> 52.111.252.7
52.111.252.7 -> 192.168.0.103
192.168.0.103 -> 192.178.24.163
192.178.24.163 -> 192.168.0.103
142.250.184.106 -> 192.168.0.103
192.168.0.103 -> 142.250.184.106
```

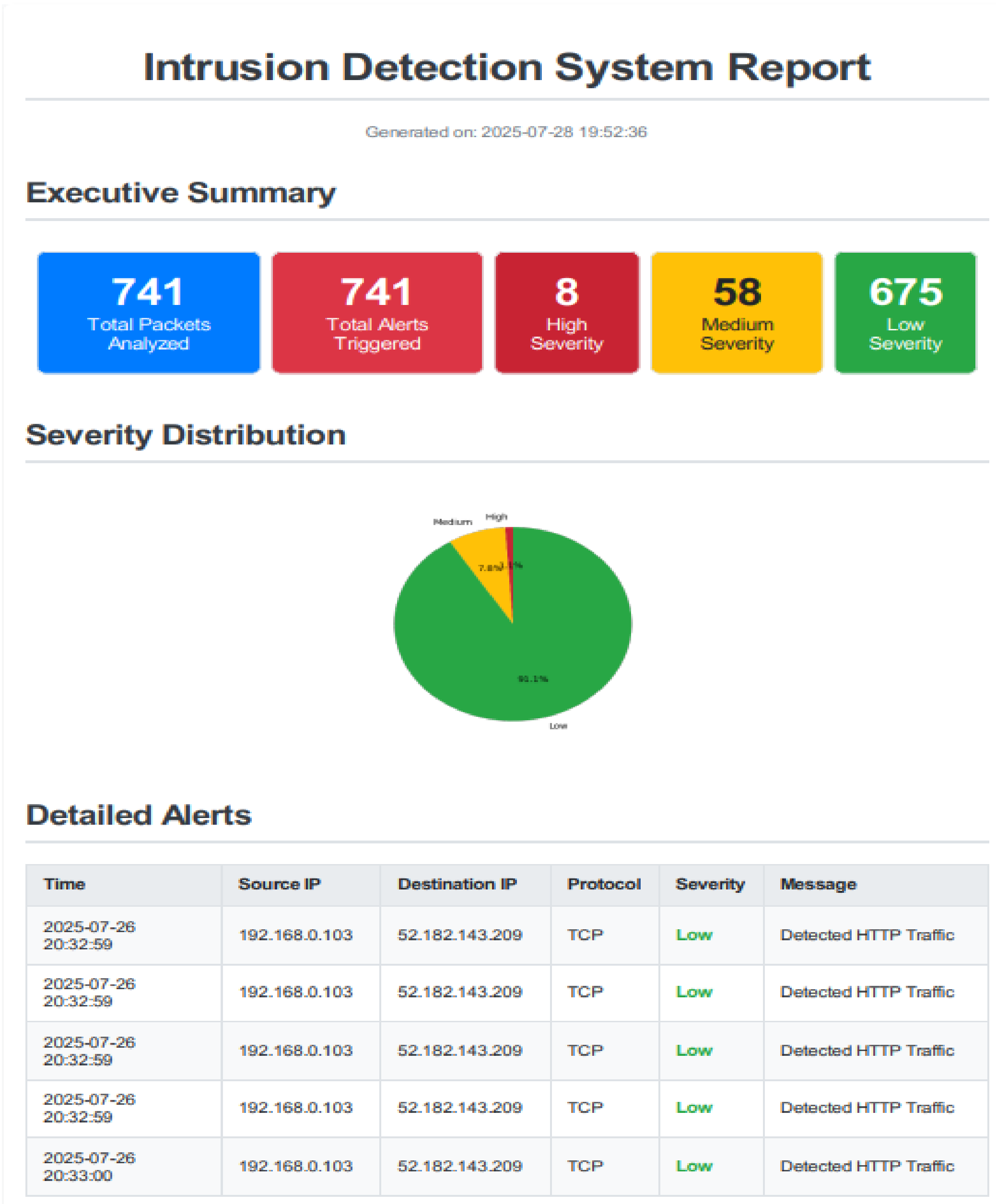
Recent Alerts

Time	Source	Destination	Protocol	Severity	Message
19:54:50	192.168.0.103	35.190.80.1	TCP	Low	Detected HTTP Traffic
19:54:50	192.168.0.103	35.190.80.1	TCP	Low	Detected HTTP Traffic
19:54:50	192.168.0.103	35.190.80.1	TCP	Low	Detected HTTP Traffic
19:54:50	192.168.0.103	35.190.80.1	TCP	Low	Detected HTTP Traffic
19:54:49	192.168.0.103	35.190.80.1	TCP	Low	Detected HTTP Traffic
19:54:49	192.168.0.103	35.190.80.1	TCP	Low	Detected HTTP Traffic
19:54:49	192.168.0.103	13.107.246.63	TCP	Low	Detected HTTP Traffic
19:54:48	192.168.0.103	13.107.246.63	TCP	Low	Detected HTTP Traffic

c. Chart of Protocol Distribution



d. Generated HTML Report



Intrusion Detection System Report

Generated on: 2025-07-28 19:52:36

Executive Summary

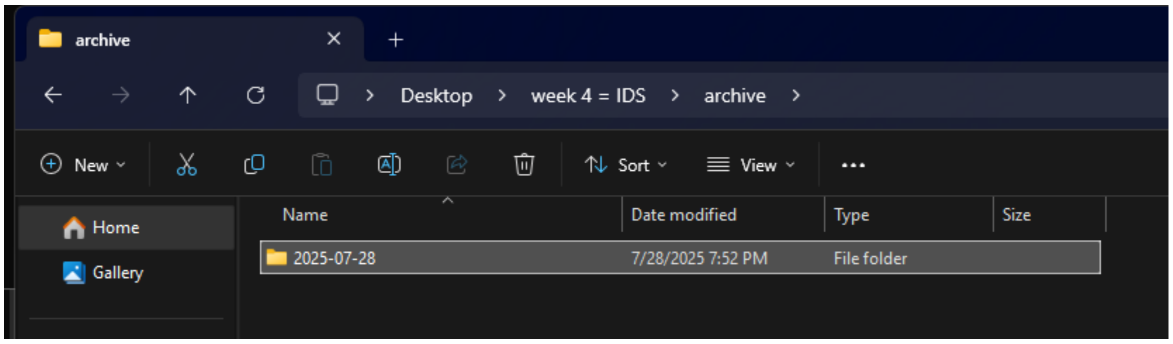
741 Total Packets Analyzed	741 Total Alerts Triggered	8 High Severity	58 Medium Severity	675 Low Severity

Severity Distribution

Detailed Alerts

Time	Source IP	Destination IP	Protocol	Severity	Message
2025-07-26 20:32:59	192.168.0.103	52.182.143.209	TCP	Low	Detected HTTP Traffic
2025-07-26 20:32:59	192.168.0.103	52.182.143.209	TCP	Low	Detected HTTP Traffic
2025-07-26 20:32:59	192.168.0.103	52.182.143.209	TCP	Low	Detected HTTP Traffic
2025-07-26 20:32:59	192.168.0.103	52.182.143.209	TCP	Low	Detected HTTP Traffic
2025-07-26 20:33:00	192.168.0.103	52.182.143.209	TCP	Low	Detected HTTP Traffic

e. Archived Logs Folder View



Name	Date modified	Type	Size
2025-07-28	7/28/2025 7:52 PM	File folder	

Appendix

Filename	**Purpose**
`main_detection.py`	Core detection logic and rule matching
`ids_gui.py`	Main GUI with dashboard and controls
`generate_report.py`	HTML and PDF report generation
`archive_logs.py`	Archive old logs into timestamped folders
`rules.txt`	Custom detection rules
`suspicious_packets.json`	Stores active alerts
`config.ini`	Configuration file (interface, archive)
`report_template.html`	Jinja2 template for HTML reports