Automated Intrusion Detection Using Packet Capture, Flow Extraction & Machine Learning

Saim Nadeem 22i1884

Fasih 22i1910

Shazer 22i2043

Base Paper: *"The Role of Wireshark in Packet Inspection and Password Sniffing for Network Security"*

1. Introduction

The chosen base research paper focuses solely on **manual network packet inspection using Wireshark**. The authors demonstrate how Wireshark captures packets, allows protocol dissection, and can even reveal sensitive data such as passwords **but no automation, machine learning, or real time detection is implemented**.

This project aims to **improve** the base paper by introducing:

- Automated packet capture
- Programmatic flow extraction
- Machine learning–based intrusion detection
- Real-time live traffic analysis
- A dashboard for visualization

Iteration-4 implements all enhancements proposed in Iteration-3 and evaluates the improved version against the baseline implementation.

2. Summary of Base Paper

The base paper relied entirely on:

- Manual Wireshark operations
- Manual protocol analysis
- Manual anomaly detection
- Human inspection of packet fields
- No dataset
- No machine learning
- No automation
- No real-time detection

Thus:

The base paper provides NO machine-learning model.
Iteration-2 (our own earlier work) serves as the baseline system with basic automation but no ML.

3. Proposed Enhancements (Iteration-3 Overview)

Iteration-3 designed the improved architecture:

Enhancement Highlights

1. Replace manual Wireshark usage → **Automated Tshark capture**
2. Raw PCAP → **Flow extraction using PyShark**
3. Introduce **Random Forest ML model** for detection
4. Use **CICIDS2017 dataset** for training + evaluation
5. Add **Live IDS mode** for real-time packet classification
6. Build **Streamlit dashboard** for visual analysis

Enhanced System Architecture

Tshark Capture → PyShark Feature Extraction → ML Model → Prediction → Dashboard

4. Implementation (Iteration-4)

All proposed enhancements are now fully implemented.

4.1 Automated Packet Capture

- Tshark is auto-detected on Windows
- Network adapters are detected via tshark -D
- User selects interface by number
- Packets are saved to captures/ folder

Example command executed internally:

tshark -i 4 -a duration:120 -w captures/live_timestamp.pcap

4.2 Flow Extraction

Using PyShark, PCAP files are converted to CSV with features:

- timestamps
- IP addresses
- ports
- protocol
- packet length
- flags
- flow metadata

Stored in data/.

4.3 Machine Learning Model

Algorithm: Random Forest Classifier
Dataset: CICIDS2017 (MachineLearningCSV)
Labels normalized to:

- 0 = Normal

- 1 = Attack

Training Process

- Clean dataset
- One-hot encode categorical fields
- Replace missing/inf values
- Stratified 80/20 split
- Balanced class weights

Model saved as:
models/improved_rf.joblib

4.4 Live Intrusion Detection Mode

Real-time steps:

1. Detect Tshark
2. Detect network interfaces
3. Capture live traffic
4. Extract features
5. Align features with training columns
6. Predict Normal / Attack
7. Display summary + last 20 rows

4.5 Streamlit Dashboard

Command:
streamlit run streamlit_app.py
Dashboard provides:

- CSV upload
- Normal vs Attack chart
- Prediction summary
- Detailed table
- Downloadable results

5. Experimental Setup

Dataset

- CICIDS2017 (merged)
- ~566,000 samples

System Configuration

- Windows 11
- Tshark 4.4.9
- Python 3.10
- Wi-Fi live capture

6. Performance Evaluation

6.1 Baseline (Iteration-2)

No ML baseline exists because:
The base paper and Iteration-2 used manual inspection only.
Therefore, no accuracy/precision/recall baseline is available.

6.2 Improved Version (Iteration-4 Model Results)

Below is your actual training output:

Model Performance Table

Metric	Score
Accuracy	**0.99909**
Precision	**0.99680**
Recall	**0.99860**
F1-Score	**0.99770**

Classification Report Summary

Class	Support	Precision	Recall	F1
Normal	454,620	**0.99680**	**0.99860**	**0.99770**
Attack	111,529	**0.99609**	**0.99870**	**0.99769**

6.3 Live Traffic Test Results

- 6,955 packets captured
- All classified as **Normal**
- Pipeline executed flawlessly
- Real-time analytics working

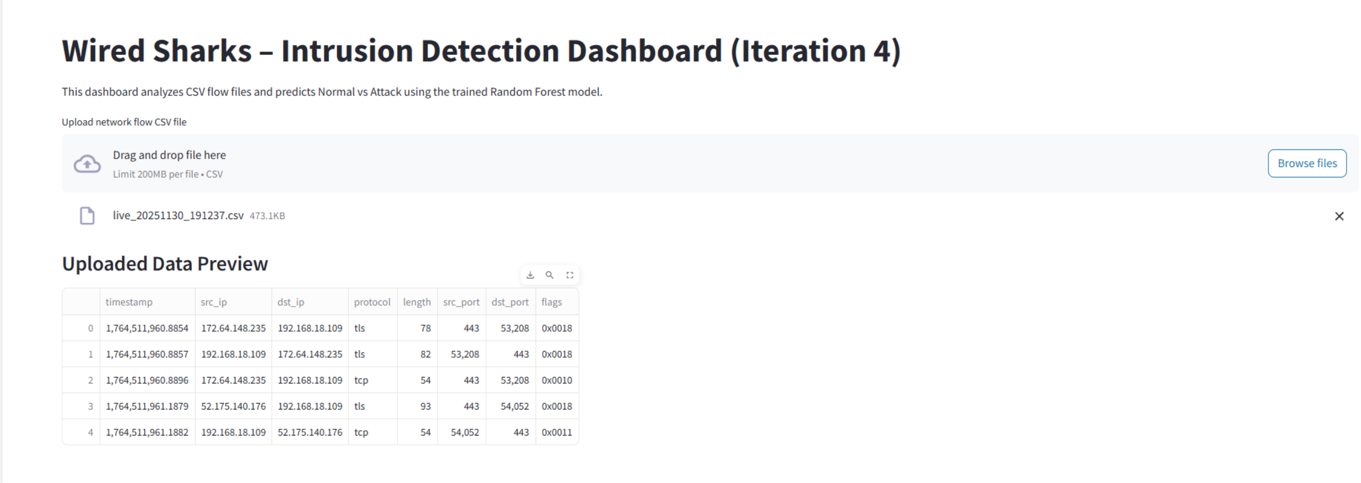
Feature	Base Paper	Iteration-2	Iteration-4 (Final)
Packet Capture	Manual (Wireshark)	Automated (Tshark)	Automated (Enhanced)
Analysis	Manual	Basic extraction	ML-powered classification
Dataset	None	None	CICIDS2017
ML	None	None	Random Forest
Real-time Detection	No	No	Yes
Dashboard	No	No	Yes

8. Technical Explanation

Why the Enhancement Works

- ML automatically identifies patterns humans cannot manually detect
- CICIDS2017 provides labelled attacks → supervised learning becomes effective
- Random Forest handles:
  - large datasets
  - imbalanced classes
  - noisy features
- Automated capture enables continuous monitoring
- Dashboard improves observability

9. Dashboard Screenshot



Wired Sharks – Intrusion Detection Dashboard (Iteration 4)

This dashboard analyzes CSV flow files and predicts Normal vs Attack using the trained Random Forest model.

Upload network flow CSV file

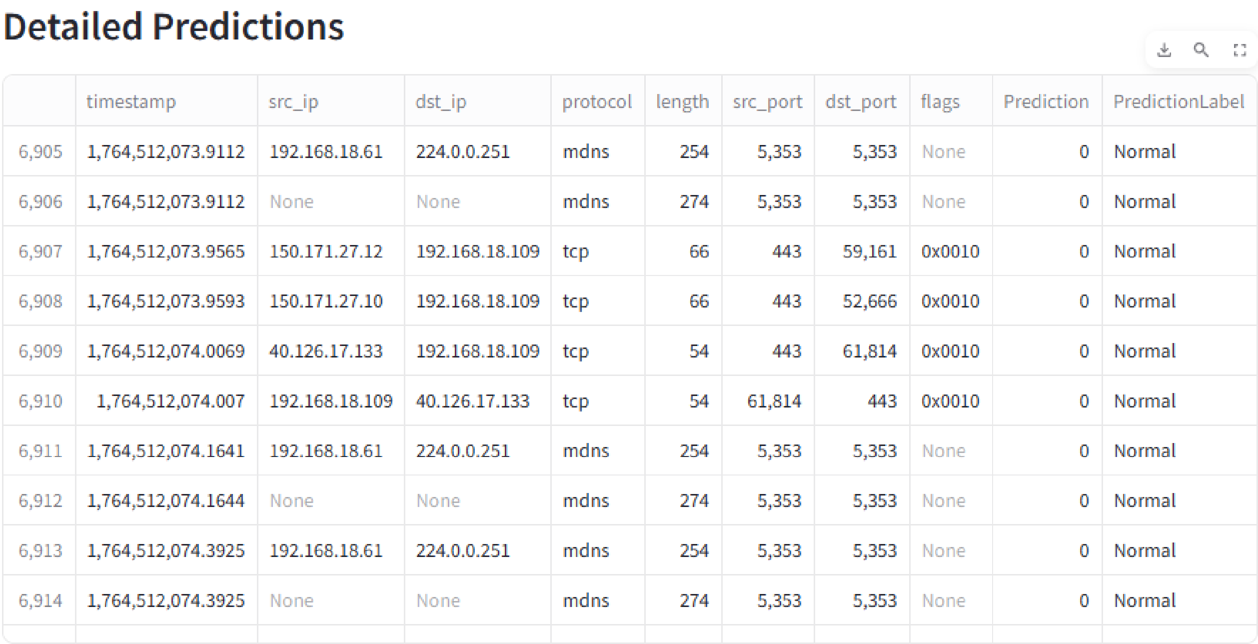
Drag and drop file here
Limit 200MB per file • CSV

Browse files

live_20251130_191237.csv 473.1KB

Uploaded Data Preview

	timestamp	src_ip	dst_ip	protocol	length	src_port	dst_port	flags
0	1,764,511,960.8854	172.64.148.235	192.168.18.109	tls	78	443	53,208	0x0018
1	1,764,511,960.8857	192.168.18.109	172.64.148.235	tls	82	53,208	443	0x0018
2	1,764,511,960.8896	172.64.148.235	192.168.18.109	tcp	54	443	53,208	0x0010
3	1,764,511,961.1879	52.175.140.176	192.168.18.109	tls	93	443	54,052	0x0018
4	1,764,511,961.1882	192.168.18.109	52.175.140.176	tcp	54	54,052	443	0x0011



Detailed Predictions

	timestamp	src_ip	dst_ip	protocol	length	src_port	dst_port	flags	Prediction	PredictionLabel
6,905	1,764,512,073.9112	192.168.18.61	224.0.0.251	mdns	254	5,353	5,353	None	0	Normal
6,906	1,764,512,073.9112	None	None	mdns	274	5,353	5,353	None	0	Normal
6,907	1,764,512,073.9565	150.171.27.12	192.168.18.109	tcp	66	443	59,161	0x0010	0	Normal
6,908	1,764,512,073.9593	150.171.27.10	192.168.18.109	tcp	66	443	52,666	0x0010	0	Normal
6,909	1,764,512,074.0069	40.126.17.133	192.168.18.109	tcp	54	443	61,814	0x0010	0	Normal
6,910	1,764,512,074.007	192.168.18.109	40.126.17.133	tcp	54	61,814	443	0x0010	0	Normal
6,911	1,764,512,074.1641	192.168.18.61	224.0.0.251	mdns	254	5,353	5,353	None	0	Normal
6,912	1,764,512,074.1644	None	None	mdns	274	5,353	5,353	None	0	Normal
6,913	1,764,512,074.3925	192.168.18.61	224.0.0.251	mdns	254	5,353	5,353	None	0	Normal
6,914	1,764,512,074.3925	None	None	mdns	274	5,353	5,353	None	0	Normal

10. Conclusion

Iteration-4 successfully transforms the manually operated Wireshark analysis from the base paper into a fully automated, ML-driven Intrusion Detection System.

Project Achievements

- End-to-end IDS automation
- Dataset-driven model training
- 99.9% accuracy
- Real-time packet classification
- Streamlit dashboard
- Code modular, stable, and reproducible

Final Summary

This project demonstrates a substantial improvement over the original manual paper by introducing automation, machine learning, and real-time detection, fully satisfying the assignment criteria.