# Phish-ELK User Manual

This guide provides a comprehensive walkthrough for setting up, using, and automating the Phish-ELK pipeline — a containerized phishing detection and response system built on the ELK Stack (Elasticsearch, Logstash, Kibana), Filebeat, and Python.

---

## 🧱 Architecture Overview

Filebeat: Harvests email logs from a mounted volume.

Logstash: Parses logs and forwards structured events.

Elasticsearch: Stores indexed phishing data.

Kibana: Visualizes phishing indicators.

Python: Automates quarantine actions via a mock API.

---

## ⚙️ Setup Instructions

### 1. Clone the GitHub Repository

# Clone your repo from GitHub

*git clone https://github.com/Anant0406/Phish-ELK.git*

### 2. Navigate to Phish-ELK Directory

*cd Phish-ELK*

### 3. Start the ELK Stack

*docker compose up -d*

## Access Kibana Dashboard

- Open your web browser and navigate to http://localhost:5601
- This URL will open the Kibana dashboard where you can visualize phishing data and interact with the ELK stack.
- Ensure Docker containers are running before accessing Kibana.

## 4. Simulate Email Logs

Inside the Filebeat container:

*echo '2025-12-31T13:00:00Z src_ip=203.0.113.55 sender=alerts@phishy.biz recipient=you@company.local subject="Verify Now" verdict=quarantined urls=http://phishy.biz/verify attachments=invoice.exe' >> /data/email_logs.log*

## 5. Confirm Ingestion

*curl http://localhost:9200/_cat/indices?v*

Look for email-phishing-YYYY.MM.DD index with non-zero docs.count.

---

## 📊 Kibana Configuration

### 1. Create Data View

Navigate to Stack Management → Data Views → Create data view

Name: email-phishing

Pattern: email-phishing-*

Time field: @timestamp

## 2. Explore in Discover

Go to Discover

Select email-phishing data view

Filter by:

verdict: quarantined

sender: alerts@phishy.biz

---

## 📊 Dashboard Visualizations

## Panels:

*Top Senders: Bar chart by sender*

*Verdicts Over Time: Line chart split by verdict*

*Quarantined Attachments: Table of attachments, subject, sender*

*Sender Domain vs Verdict: Heatmap of sender.keyword vs verdict*

*Phishing URLs (Pie Chart): Shows most frequent phishing URLs and quarantine status*

## Import Dashboard

*Go to Stack Management → Saved Objects → Import*

*Import:* ***email-phishing-dashboard.ndjson***

---

## 🤖 Python Automation

### **Script**: quarantine.py

Queries Elasticsearch for verdict=quarantined

Sends metadata to mock API at http://localhost:5000/quarantine

### Run Script

*python3 python/quarantine.py*

### Expected Output

Found 3 quarantined emails.

Quarantined: Verify Now → Status: 200

Quarantined: Statement → Status: 200

---

🦤 **Mock API Setup:** Has to be done before running the python script

### **Script**: mock_api.py

```
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route("/quarantine", methods=["POST"])

def quarantine():

    data = request.get_json()

    print(f"Quarantined email: {data}")

    return jsonify({"status": "success"}), 200
```

app.run(host="0.0.0.0", port=5000)

## Run API

*python3 mock_api.py*

---

## 📦 Requirements

*flask*

*requests*

*Install Dependencies*

---

## 📁 Deliverables

*filebeat.yml, phishing.conf*

*quarantine.py, mock_api.py*

*email-phishing-dashboard.ndjson,*

*user-guide.pdf,*

*requirements.txt*

---

## 📓 Notes

Ensure Docker volumes are mounted correctly.

Use localhost consistently across services.
Run Flask and Python scripts in the same environment (WSL or Windows).

---