**PRIVACY PRESERVING NETWORK ANALYSIS USING HOMOMORPHIC ENCRYPTION**

**Group Members:**

Shardul Shekatkar – 1032222332

Keyuar Bhelonde – 1032222651

Harsh Zavare – 1032222462

Jesal Shah – 1032230403

Parth Rai – 1032222464

Navdeep Bhargava – 1032222736

**Privacy-Preserving Network Analysis:**

**Objectives**

The primary objectives of this project are:

1. **Secure Packet Capture and Processing** – Capture network packets securely while minimizing privacy risks.
2. **Homomorphic Encryption for Privacy-Preserving Analysis** – Utilize CKKS homomorphic encryption (via TenSEAL) to perform encrypted computations on network traffic data without decrypting it.
3. **Encrypted Anomaly Detection** – Implement encrypted machine learning techniques to detect network anomalies while preserving data privacy.
4. **Statistical and ML-Based Anomaly Detection** – Use statistical methods (mean, standard deviation) and machine learning (Isolation Forest) for anomaly detection.
5. **Data Visualization and Reporting** – Generate graphical representations (histograms) and structured reports of packet statistics and anomalies.

**Dataset used:**

**NSL-KDD Dataset:**

**(Reference Dataset for ML Model Training in the Provided Code)**

**1. What is the NSL-KDD Dataset?**

- A benchmark dataset for network intrusion detection systems (IDS).

- An improved version of the older KDD Cup 1999 dataset, addressing its limitations (e.g., redundant records, bias).

- Contains simulated network traffic with labeled attacks and normal connections.

**2. Key Features of NSL-KDD:**

|  |  |
| --- | --- |
| **Feature** | **Details** |
| **Records** | **~148,517 total (train + test)** |
| **Attack Types** | **DoS, Probe, R2L, U2R** |
| **Features** | **41 features per record (e.g., duration, src\_bytes, protocol\_type)** |
| **Labels** | **Binary (normal or attack) + attack-type labels** |
| **Purpose** | **Train/test ML models for anomaly-based intrusion detection** |

**3. Dataset Structure (As Used in the Code):**

**The code extracts 3 numerical features (Line 242):**

features = data[['duration', 'src\_bytes', 'dst\_bytes']]

duration: Connection length (seconds).

src\_bytes: Bytes sent from source to destination.

dst\_bytes: Bytes sent from destination to source.

(Note: The full dataset has 38+ additional features, but the code simplifies it for demonstration.)

**4. How the Code Uses NSL-KDD:**

**Loading Data (Line 239):**

data = pd.read\_csv("nsl\_kdd.csv", header=None, names=columns)

Loads the dataset with predefined column names.

**Training ML Model (Lines 256–259):**

model = IsolationForest(contamination=0.1)

model.fit(features) # Uses the 3 numerical features

Trains an Isolation Forest (unsupervised anomaly detection).

**Performance Report (Lines 288–294):**

Logs anomaly detection metrics (e.g., total samples, attack rate).

**Limitations in the Code’s Implementation:**

* Simplified Features: Only 3/41 features are used (for demo purposes).
* No Label Utilization: The code ignores attack-type labels (unsupervised approach).
* Legacy Use: The NSL-KDD model isn’t applied to encrypted data (live packets use a custom method).

**Model Accuracy:**

**1. For Statistical Anomaly Detection**

*(Used in detect\_anomalies() - Line 124)*

* Method: Flags packets outside mean ± 2.0 \* std\_dev.
* Estimated Accuracy:
  + ~95% for normal traffic (assuming Gaussian distribution).
  + False Alarms: Likely high for non-Gaussian traffic (e.g., video streams).

Example:

* If mean = 500 bytes, std\_dev = 100:
  + Packets < 300 bytes or > 700 bytes are flagged.
  + 5% of normal packets may be mislabeled as anomalies.

**2. For Encrypted ML Detection**

*(Used in encrypted\_ml\_anomaly\_detection() - Line 157)*

* Method: Scores packets with 0.5\*x + 0.01\*x² - 5.0.
* Estimated Accuracy:
  + ~70-80% for simple attacks (e.g., DoS with large packets).
  + Fails for subtle attacks (e.g., slow port scans).

**Example:**

* A 1000-byte packet gets score = 0.5\*1000 + 0.01\*1000000 - 5.0 = 500 + 10000 - 5 = 10,495.
  + If threshold = 0: Flagged as anomaly (likely correct for DoS).

**3. For NSL-KDD Isolation Forest**

*(Used in train\_ml\_model\_from\_dataset() - Line 256)*

* Method: Unsupervised clustering.
* Estimated Accuracy:
  + ~85-90% on NSL-KDD test data (for known attacks).
  + Drops to ~60% for new attack types.

Example:

* On 100 packets with 10 true attacks:
  + Detects 8-9 attacks but may miss 1-2 (false negatives).

**Implementation Status:**

| **Objective** | **Implementation Status** | **Details** |
| --- | --- | --- |
| Secure Packet Capture and Processing | ✅ Implemented | Scapy is used to capture live network packets securely. |
| Homomorphic Encryption for Privacy-Preserving Analysis | ✅ Implemented | CKKS encryption (via TenSEAL) is applied to packet lengths before analysis. |
| Encrypted Anomaly Detection | ✅ Implemented | A privacy-preserving anomaly detection mechanism using encrypted arithmetic is developed. |
| Statistical and ML-Based Anomaly Detection | ✅ Implemented | Both statistical (threshold-based) and ML-based (Isolation Forest, encrypted model) methods are included. |
| Data Visualization and Reporting | ✅ Implemented | Packet histograms are generated, and analysis reports are exported in CSV format. |

**A screenshot of a computer

AI-generated content may be incorrect.**

**CSV REPORT**

**A screenshot of a computer

AI-generated content may be incorrect.**

**VISUALIZATION**

**A screen shot of a graph

AI-generated content may be incorrect.**

**MODEL PERFORMACE DETAILS**

**A screenshot of a computer

AI-generated content may be incorrect.**

**DETECTED ANOMALIES**

**A screenshot of a computer

AI-generated content may be incorrect.**