User Manual : Supervised Flow Classification

Introduction

This manual explains how to use the provided script to classify network flows from PCAP files and detect anomalies using supervised methods. The script processes PCAP files, extracts flow features, enriches data with advanced metrics, and labels flows based on ground truth data.

---

1. Prerequisites

Before using the script, ensure you have the following:

- **PCAP Files:** Store these in a directory named `pcap_files`.
- **Ground Truth File:** A CSV file named `TRAIN.gt` that lists relevant flows for labeling.
- **Dependencies Installed:** Install required Python libraries:
  - `nfstream`, `pandas`, `numpy`, `scikit-learn`, `scapy`, `tqdm`.

Use the following command to install missing packages:

```
pip install nfstream pandas numpy scikit-learn scapy tqdm
```

---

2. How to Run the Script

2.1 Setting Up

1. Place your PCAP files in the `pcap_files` directory.
2. Ensure the ground truth file (`TRAIN.gt`) is in the `pcap_files/flows/` directory.

2.2 Execution

Run the notebook in colab or jupyter :

```
1_2_data_preparation.ipynb
```

2.3 Output Files

After execution, the following files will be generated:

- `csv_files/raw_flows.csv`: Raw extracted flows with basic metrics.
- `csv_files/enriched_flows.csv`: Enriched flows with fan-in, fan-out, and other features.
- `csv_files/labeled_flows.csv`: Flows labeled as normal (0) or anomalous (1).
- `csv_files/final_features_flows.csv`: Final dataset with advanced features.

---

3. Detailed Workflow

3.1 Processing PCAP Files

The script automatically detects and processes all `.pcap` files in the `pcap_files` directory. Each file is analyzed using `NFStreamer`, configured with:

- **Idle Timeout:** 60 seconds.
- **Active Timeout:** 120 seconds.
- **Statistical Analysis:** Enabled for metrics like mean and standard deviation of packet sizes.

Progress is displayed in the terminal for each file.

3.2 Enriching Data

Fan-In and Fan-Out Metrics

The script calculates:

- **Fan-In:** Number of unique IPs connecting to a specific IP.
- **Fan-Out:** Number of unique IPs connected by a specific IP. A sliding window of 10 seconds (`T=10`) is used for these calculations.

IP Address Classification

Each IP is categorized into classes (A, B, C, D, E) based on its first octet. This classification is encoded into one-hot vectors for analysis.

Normalization

Numerical columns such as packet size and duration are normalized using `StandardScaler` for consistent scaling.

3.3 Labeling the Data

Flows are matched against the ground truth (`TRAIN.gt`) to determine:

- `1`: Anomalous flows.
- `0`: Normal flows.

The labeling process matches flows by comparing:

- Source and destination ports.
- Protocols.
- Timestamp overlaps between the flows and ground truth entries.

3.4 Advanced Feature Engineering

The script generates additional metrics to improve anomaly detection:

1. **Connection Patterns:**
   - Fan ratios, connection asymmetry, and connectivity intensity.
2. **Timing Features:**
   - Ratios and regularity of packet inter-arrival times.
3. **Protocol and Port Analysis:**
   - Detects suspicious combinations and anomalies.
4. **Packet Characteristics:**
   - Packet size ratios, flow efficiency, and regularity.
5. **Behavioral Features:**
   - Detects scans, DDoS attempts, and potential data exfiltration.

All features are combined into a comprehensive dataset (`csv_files/final_features_flows.csv`).

---

4. Understanding the Outputs

4.1 Raw Flows

The file `raw_flows.csv` contains basic extracted metrics such as:

- Source/Destination IPs.
- Ports and protocol.
- Packet size statistics.

4.2 Enriched Flows

The file `enriched_flows.csv` includes additional metrics:

- Fan-In and Fan-Out.
- IP classifications.
- Duration and packet inter-arrival statistics.

4.3 Labeled Flows

The file `labeled_flows.csv` contains the same data as `enriched_flows.csv` with an additional `label` column:

- `1` for anomalous flows.
- `0` for normal flows.

4.4 Final Dataset

The file `final_features_flows.csv` is the fully enriched dataset:

- Combines all extracted features.
- Ready for machine learning tasks such as training classifiers.

---

5. Customization Options

Adjusting the Time Window

To change the time window (`T`) for Fan-In/Fan-Out calculations:

1. Locate the line:

   ```
   T = 10  # 10 seconds window
   ```

2. Update the value of `T` as desired.

Adding New Features

The script includes modular functions to add new features:

- **Connection patterns:** `create_connection_pattern_features()`
- **Timing features:** `create_timing_features()`
- **Protocol analysis:** `create_protocol_features()`
- **Packet analysis:** `create_packet_features()`
- **Behavioral detection:** `create_behavioral_features()`

Modify or extend these functions to include additional metrics.

---

6. Troubleshooting

Error: Missing PCAP Files

- Ensure PCAP files are in the `pcap_files` directory.
- Verify filenames end with `.pcap`.

Error: Missing Ground Truth File

- Place `TRAIN.gt` in `pcap_files/flows/`.

Error: Missing Libraries

- Install dependencies using:

  ```
  pip install -r requirements.txt
  ```

---

7. Additional Notes

- **Performance Considerations:**
  - The script uses efficient processing with Pandas and vectorized operations.
  - Progress bars (`tqdm`) provide real-time feedback for large datasets.

- **Extensibility:**
    - The script can be integrated into a larger pipeline for cybersecurity monitoring and anomaly detection.

For questions or issues, consult the script comments or modify the code to suit specific requirements.