Generative Adversarial Network for Malicious and Benign Traffic Generation

Project Description

Introduction

As cyber threats become increasingly complex, the ability to generate realistic network traffic — both benign and malicious — is crucial for advancing cyber-security systems. Traditional simulation methods often fall short in emulating the diversity and complexity of real-world traffic. To address this, we propose a machine learning-based approach using **Generative Adversarial Networks (GANs)**, a form of generative AI, to synthesize realistic traffic patterns.

What is a Neural Network?

A neural network is a computational model inspired by the human brain. It consists of layers of interconnected nodes (neurons), where each connection has a weight that is adjusted during training. Neural networks are widely used in pattern recognition, classification, and regression tasks. They are capable of learning complex relationships in data through backpropagation and gradient descent.

What is a Generative Adversarial Network (GAN)?

A Generative Adversarial Network (GAN) is a type of neural network architecture introduced by Ian Goodfellow in 2014. It consists of two main components:

- **Generator (G)**: Attempts to generate synthetic data that mimics the real data distribution.
- **Discriminator (D)**: Tries to distinguish between real data and data generated by the Generator.

Both components are trained in a minimax game: the Generator improves to fool the Discriminator, while the Discriminator learns to detect fake data. Over time, this adversarial process helps the Generator produce highly realistic samples.

Project Task

The objective of this project is to train a GAN model on a labeled network traffic dataset that includes both benign and malicious records using **PyTorch**. The GAN will then be used to generate synthetic traffic samples that can be used to augment existing datasets for training Intrusion Detection Systems (IDS), or to test their robustness.

Steps Involved

1. **Preprocessing the Data**
    - *Data Set Size*: You can use part of the dataset, at least 100,000 entry randomly sampled.
    - *IP Address Handling*: Treat IP addresses as numerical data by splitting each IP into four separate columns (octets), e.g., for 192.168.1.1: create columns IP_1=192, IP 2=168, IP 3=1, IP 4=1.
    - *Categorical Features*: Apply label encoding or one-hot encoding to categorical fields like protocol, service, or flag.
    - *Numerical Features*: Scale all numerical features (e.g., byte counts, duration, etc.) using standardization (Z-score) or min-max normalization.
2. **Training the GAN**
    - Train the Discriminator to distinguish between real and fake traffic samples.
    - Train the Generator to create synthetic samples that resemble real data distributions.
3. **Traffic Generation**
    - Use the trained Generator to produce new malicious and benign samples.
    - Make Sure that the Generated Sample are in Human Readable Format (The Labels and ip addresses and so on)

Expected Outcomes

- A trained GAN capable of generating realistic labeled traffic samples.
- A pipeline for preprocessing raw network traffic into a format suitable for neural network training.
- Insights into the strengths and weaknesses of generative models for cybersecurity applications.