BotNet Detection using ML.

Packages Used:

1. Scapy (for parsing pcap files):
   ```
   pip install --pre scapy[complete]
   ```

2. Numpy:
   ```
   pip install numpy
   ```

3. Pandas:
   ```
   pip install pandas
   ```

4. Scikit-learn:
   ```
   pip install scikit-learn
   ```

Submission Files:

FeatureExtractor.py: Contains the class which is used to generate 25 aggregate features from a list of packets for a flow, the features were decided by looking up common ways to detect botnets and ddos.

Preprocessing.py: Used to read the dataset and generate a csv file which will be used for training, it uses the FeatureCalc class from FeatureExtractor. It outputs a csv file “Botnet_train.csv

Training.py: Contains the code used to train the model, it reads the Botnet_train.csv file and trains the model, finds the metrics also.

Botnetdetect.py: The submission file which takes a pcap file as input and outputs out.txt which has information about traffic flows being malicious/botnet. ***It only outputs a flow if it is malicious. Doesn't output Benign flow of packets within the network***

Models Directory which contains the trained model, we trained a random forest classifier, it also contains a fit scaler for scaling the data.

Botnet_train.csv: The extracted training dataset.

Notebook used for training : Colab Notebook

Note:

I had to do some Undersampling to make sure the model was not biased as the number of benign traffic samples outweighed the botnets in the training set.

Results/Metrics:

- The model has very high accuracy on the test samples, for all the times the model was trained, the accuracy was always pretty high.

Accuracy = 0.9997124311888916

- The model has amazing precision recall, f1 scores which indicate that it has a very low number of false positives and false negatives.

Precision = 0.9997124311888916 Recall = 0.9869706840390879
F1 = 0.9908436886854153

- All these metrics can be verified by running the **colab notebook** linked above (after mounting google drive with the dataset).