README

Running Instructions –

```
python botnetdetect.py <absolute_path_for_pcap_file>
```

This will create a file "*result.txt*" in the directory where the script executes. It will also generate a csv called "*flows_preprocessed_with_prediction.csv*", that shows classification of every flow detected in the test PCAP file as benign or malicious.

Other Deliverables –

- ➔ All the pickled models are present in the "*Models"* Directory.
- ➔ Folder wise and total pre-processed data is present in the "*Training_files/pre_processed_csv"* Directory.
- ➔ All the Source code for model training is present in the "*Training_data*".
  - Running model.py will take the pre-processed csv files and retrain all the models and output the training and testing information.
  - Running parser.py will take a directory and use all its directory-wise PCAP files to generate a csv of traffic flow features per directory.
- ➔ A documentation pdf named "*Tool Documentation*" is also submitted. It contains details about the various details of the tool.

Testing and Evaluation –

Upon running botnetdetect.py with a test PCAP file, the tool generates a result.txt file in the current directory. resut.txt file labels each unique host in the PCAP file as benign or malicious.

The structure of result.txt is-

<Host IP Address 1>, <Benign/Malicious>

<Host IP Address 2>, <Benign/Malicious>

<Host IP Address 3>, <Benign/Malicious>

Submission Directory Structure –

```
Deliverables/
├── models/
│   ├── ...
│   └── Pickled Models
│
├── Training_data/
│   │
│   ├── model.py
│   ├── parser.py
│   └── pre_processed_csv
│       ├── ...
│       └── Folderwise pre-processed CSV files
│
├── botnetdetect.py
│
├── Tool_Documentation.pdf
│
└── README.pdf
```

Solution Summary

We have used traffic flow for analysis which is defined by a 5-tuple of (source IP, Destination IP, Source Port, Destination Port, Protocol). The flows originating from the host IP have been given the label of the host, i.e. for data captured on a Benign system, only the flows originating from the benign system are labelled benign, rest flows are not considered/ignored. The flows have been filtered using port, length, duration and protocol data. We have then extracted 22 features from each of the filtered network traffic flow. We have also done correlation and PCA loading based feature ranking. Top 16 features were chosen after analysis. Next a KNN based clustering has been performed to identify benign and malicious clusters and to remove the obvious benign flows. Finally, a boosted random forest is used for classifying the remainder of the flows into malicious and benign. The result of this flow classification is used to score every host, based on the percentage of malicious flows generated by the host. A threshold of 1% is set to flag the hosts as malicious, i.e. if more than 1% of flows originated by a host are malicious, then the host is classified malicious.

Requirements

sklearn == 0.22.1

numpy == 1.16.3

os

glob

pickle == 4.0

joblib == 0.14.1

csv == 1.0

natsort

pandas == 0.25.3

scapy

matplotlib

seaborn

Other Assumptions –

- ➔ The result.txt contains pairs of IP and Label. Each unique IP in the test file is labelled by the tool. Thus, the tool classifies traffic as malicious or benign.
- ➔ The flow-wise classification is also shown in flows_preprocessed_with_prediction.csv file.
- ➔ The runtime on the current system with i7 @ 1.8 GHz and 16 GB RAM is 4-5 minutes for a 100 Mb PCAP file. Please ensure appropriate time is given based on testing system specs.