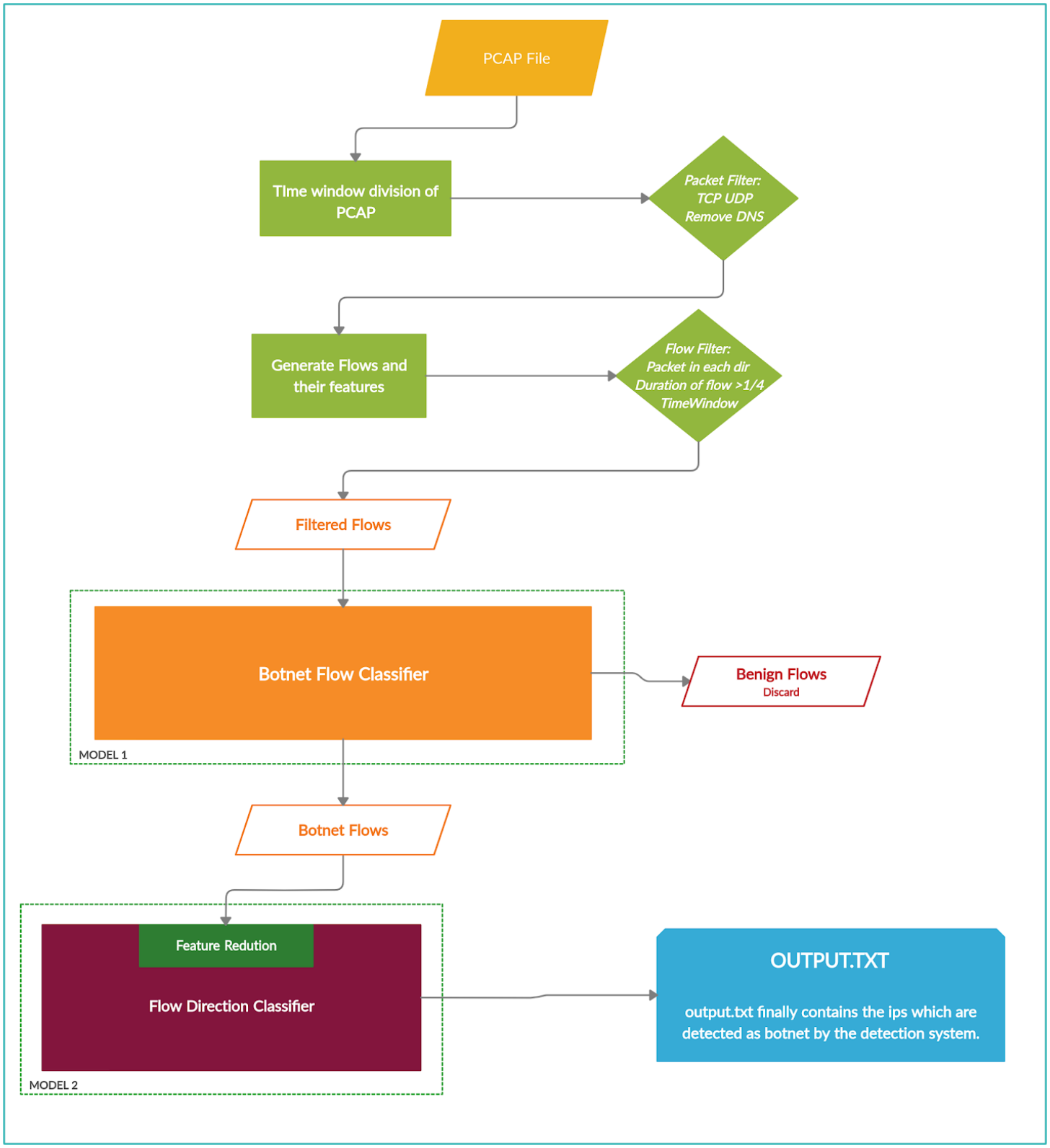
BOTNET DETECTION TOOL WORKING



OBSERVATIONS ABOUT DATASET :

1. The dataset had only p2p data and the IPs mentioned in the respective lists were present in all packets. We did find a handful of packets in one of the storm files which were not a part of the p2p data, but their number was insignificant compared to the total number of packets in the file.
   This enabled us to directly start working on bot-net detection, rather than trying to first classify the data as p2p and then working on the botnet detection

2. Few files for the "zeus" dataset had too few packets in them.

3. "Vinchuca" dataset also had a small amount of packet capture.

BOTNET FLOW CLASSIFICATION:

1. As there were a huge number of packets with no individual significance, we focused on **grouping these packets by flows** and calculated features for these groups.
   So, every feature vector is being calculated per flow detected.

2. **TIME WINDOW:** For every pcap file, we have calculated flows by divind the pcap in 1 hr time windows.

3. For every packet we parsed following for flow feature calculation:
   a. Source and destination IP and ports + Protocol (to define a flow)
   b. Packet size with and without header
   c. Time

4. Filtering:
   a. Packets Filtered: As we detected that mostly we have received p2p packet capture files, so not much packet filtering is done. We have considered tcp, udp and non-dns packets from the pcap files provided.
   b. **Filtering for "Flows"**: We have taken into account features from flows which satisfy the following conditions:
      i. Minimum one packet in forward and backward direction
      ii. Flow duration > ¼*(Time Window)

5. **FEATURES**: The following features have been calculated for each flow. Feature selection has been done keeping in mind how on what measures a botnet traffic may differ from a benign p2p traffic. We read multiple texts available on the internet for domain knowledge and feature selection.
   a. Time based features:
      Firstly, we have considered features about flow duration and inter-arrival time statistics (mean, min, max).
      Also, we considered similar statistical features for directional inter-arrival times.
   b. Packet Nos features:
      We have taken into account the packets sent in each direction and their frequency.
      We also took into account the number of packets with large (>250) data length

and ones with smaller (<50) in each (forward and backward) direction. These features were discarded while testing as this data was coming in account from the data based features.

c. Data based features:
Similar to time based features, we took in account the total data sent in each direction in flow and statical data features (min, max, mean) for forward and backward directions.

6. FEATURE REDUCTION: Initially we had obtained 20 features for each flow. Firstly we reduced 2 features as mentioned in 5b above.
    a. We used information gain of each feature to figure out which are the useful ones.
    b. This way, we found we can further reduce 5 features as their information gain was too small as compared to other features.
    c. But these features have not been removed at this step as these were major contributors in FlowDirectionClassifier. (described in the next section)

7. Model - We used a Random Forest Classifier for this task (we name this classifier as BotnetFlowClassifier). As Random Forest Classifier follows entropy based generation, using Information Gain for feature selection makes sense.

FLOW DIRECTION CLASSIFIER:

1. Botnet Flow Classifier outputs labels for each flow, i.e. we know if a flow (which has an IP pair) is botnet traffic or not.
But the problem now is that one can't tell if the source (considered randomly in flow) is an infected host (bot) or the destination.

2. In order to eliminate this ambiguity, we trained a "flow direction classifier" which, given a botnet flow, classifies whether the source IP is infected (botnet) or whether the destination IP is infected (botnet).

3. This can be detected using the direction based features from the feature vector of a flow. So we only use 10 features to train this classifier.

4. Model - We used a Random Forest Classifier for this purpose.

5. Feature Reduction -
    a. Again, here we used information gain of each feature to figure out the useful ones. As Random Forest Classifier follows entropy based generation, using Information Gain for feature selection makes sense.
    b. We found that out of 10, only 4 were the major contributions, and the remaining 6 features had very small information gain as compared to the others.
    c. So here we only used the 4 major features to train our FlowDirectionClassifier

6. Features - The final 4 features that we use to train this classifier are:
    a. Number of bytes sent in flow
    b. Number of bytes received in flow
    c. Minimum inter-time between packets sent in forward direction
    d. Minimum inter-time between packets sent in backward direction

7. Only the flows classified as “botnet” by our BotnetFlowClassifier goes through the FlowDirectionClassifier.

PCAP PROCESSING IN BOTNETDETECT.PY TOOL:

1. To give an inference on a PCAP file, in the first stage we find out the features of the flows in a manner similar to the one described in the previous section.
2. Once the features of the different flows are found, we classify every flow as botnet or benign.
3. If a flow is classified as a botnet flow, we will find out which one of it’s IP (source or destination or both) are part of the botnet. We find the frequency of the IPs which are classified as botnets by this classifier over all the flows. With this, we get a list of hosts which are botnets with a high chance.
4. To clip false positives, we did a statistical analysis of the frequencies of IPs classified as botnets and removed the hosts which had an insignificant proportion of flows classified as botnets. This was done because there are a large number of IPs in the pcap files which are not of our significance (mostly non-P2P hosts) and these hosts only comprise a very small fraction of the actual ‘botnet’ traffic and thus can be eliminated.