# Course Project - Report

## BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection

**Objective:-** The objective of BotMiner is to **detect groups of compromised machines within a monitored network that are part of a botnet**. We do so by passively analyzing network traffic in the monitored network.

## Implementation of project:-

- Data Extraction :- Extracted records of network flows from given data.

  1. From each of the given network record extracted these features ['DestIP', 'Dport', 'Protocol', 'number_of_flows','total_size_of_flows_orig', 'total_size_of_flows_resp', 'inbound_pckts', 'outbound_pckts', 'url_path_length', 'number_of_URL_query_parameters', 'filename_length','number_of_downloaded_bytes', 'number_of_uploaded_bytes','#Src_IP','goal']
  2. Removed records which have missing fields and then normalised the data.

- Getting hosts from data :-
  1. Each host is a Machine in the monitored network.
  2. Each host is associated with a unique IP address. DestIP in the communication flow represents a hostIP
  3. There are total 538 hosts in the given data.
  4. Each host can occur in many communication flows.

- C-Plane Clustering :- Clustering network flows and records information on who is talking to whom.
  1. C-Plane monitors communication patterns(Extracted Data). Considered each network record as a communication flow.
  2. Clusterised the communication flows using X-Means algorithm.
  3. For each host maintained a list of which cluster it belongs to.

- A-Plane Clustering :- Clustering information on who is doing what.
  1. Clustering is based on what type of suspicious activity each host is doing.
  2. Goal in the above data represent the suspicious activity.
  3. Clusterised the hosts based on whether a host is a normal , backdoor or ransomware.
  4. For every host maintained a list of what suspicious activities it is doing.

our detection framework clusters similar communication activities in the C-plane (C&C communication traffic), clusters similar malicious activities in the A-plane (activity traffic), and performs cross cluster correlation to identify the hosts that share both similar communication patterns and similar malicious activity patterns. These hosts are bots in the monitored network. These bots are grouped together based on similarity.

- Cross-plane Correlation :- Once we obtain the clustering results from A-plane (activities patterns) and C-plane (communication patterns), we perform cross-plane correlation. The idea is to crosscheck clusters in the two planes to find out intersections that reinforce evidence of a host being part of a botnet.

Step1 :- filter out hosts on which we have witnessed at least one kind of suspicious activity.

Step2 :- assign score to each host h of the above filtered hosts.

$$s(h) = \sum_{\substack{i,j \\ j>i \\ t(A_i)\neq t(A_j)}} w(A_i)w(A_j)\frac{|A_i \cap A_j|}{|A_i \cup A_j|} + \sum_{i,k} w(A_i)\frac{|A_i \cap C_k|}{|A_i \cup C_k|}$$

where A = {A1,A2…Am} , C = {C1,C2…Cn} are Aplane and Cplane clusters to which h belongs.

Step3 :- filter out hosts which has score less than threshold theta.

Step 4 :- h is host that belong to set of hosts that are filtered above using score

Let A = {Ai}i=1..mB be the set of A-clusters that each contains at least one bot h ∈ B,
C = {Ci}i=1..nB be the set of C-clusters that each contains at least one bot h ∈ B.
Also, let K = A∪ C be an ordered union/set of A- and C-clusters.

We then describe each bot h ∈ B as a binary vector b(h) ∈ {0, 1}^{|K|} ,
whereby the i-th element bi = 1 if h ∈ Ki , and bi = 0 otherwise.

Step 5 :- Between every two hosts we calculate similarity as follows

$$sim(h_i, h_j) = \sum_{k=1}^{m_B} I(b_k^{(i)} = b_k^{(j)}) + I(\sum_{k=m_B+1}^{m_B+n_B} I(b_k^{(i)} = b_k^{(j)}) \geq 1)$$

where we use b(i) = b(hi) and b(j) = b(hj ), for the sake of brevity. I(X) is the indication function, which equals to one when the boolean argument X is true, and equals to zero when X is false

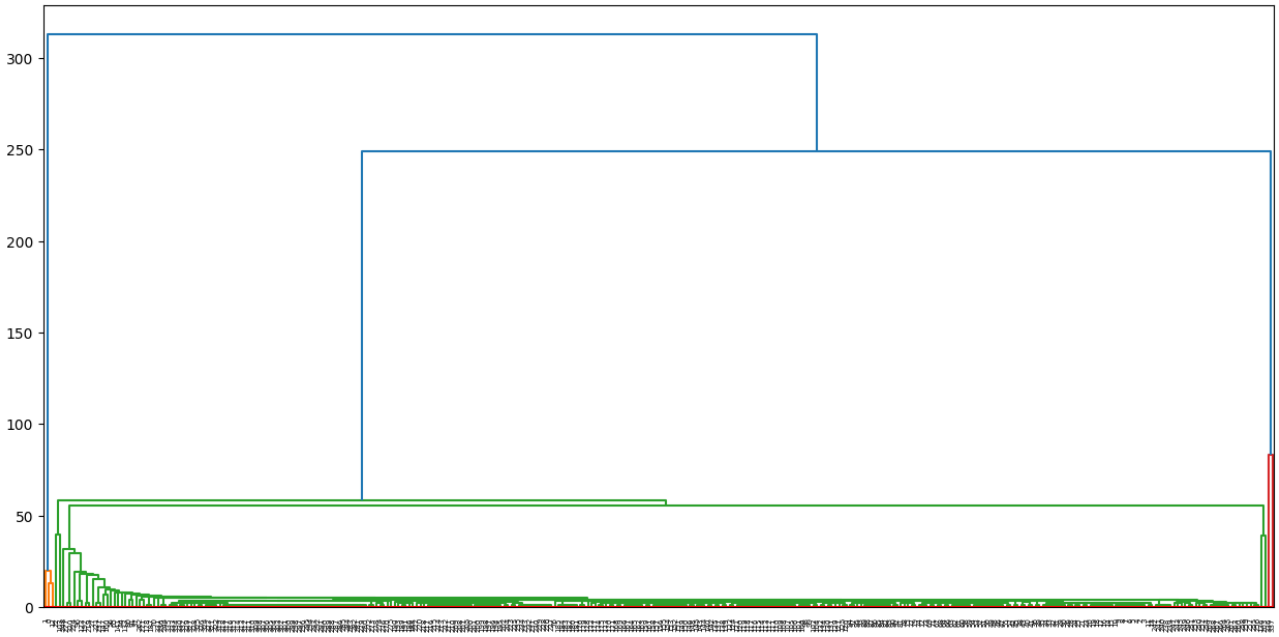
Step 6 :- This definition of similarity between hosts gives the opportunity to apply hierarchicay clustering. This allows us to build a dendrogram, i.e., a tree like graph that encodes the relationships among the bots.

Step 7 :- We use the Davies-Bouldin (DB) validation index to find the best dendrogram cut, produces the most compact and well separated clusters.
The obtained clusters group bots in (sub-) botnets

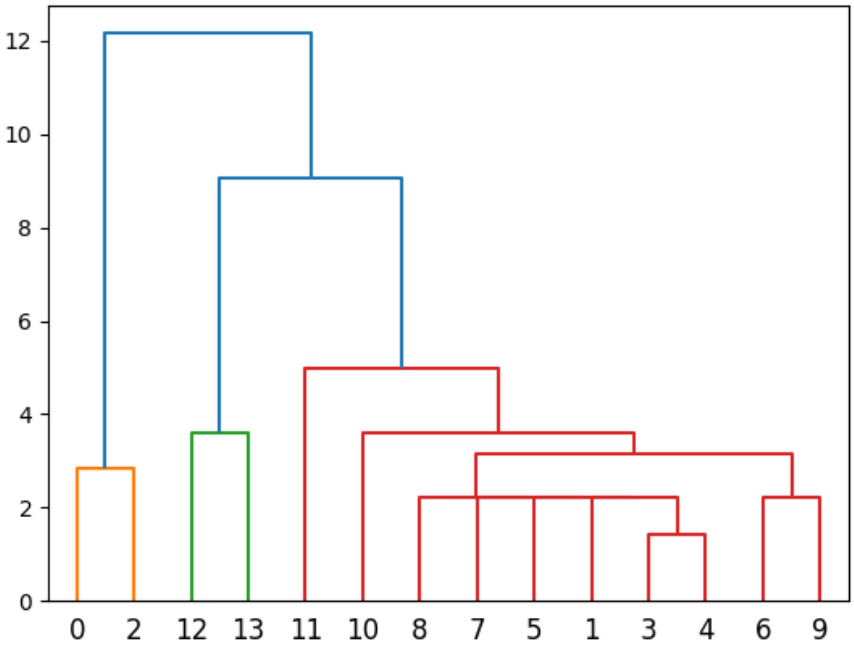
## Inferences and Results:-

- Dendogram obtained :-

1) With very low threshold




2) With high threshold




DB index is 0 in both cases which is implying that there are no botnet groups in the networks. (But single compromised machines(bots) may exist which is not objective of this paper).