# CluSiBotHealer: Botnet Detection through Similarity Analysis of Clusters

Pijush Barthakur, Manoj Dahal, and Mrinal Kanti Ghose

***Abstract*—Botnets are responsible for most of the security threats in the Internet. Botnet attacks often leverage on their coordinated structures among bots spread over a vast geographical area. In this paper, we propose CluSiBotHealer, a novel framework for detection of Peer-to-Peer (P2P) botnets through data mining technique. P2P botnets are more resilient structure of botnets (re)designed to overcome single point of failure of centralized botnets. Our proposed system is based on clustering of C&C flows within a monitored network for suspected bots. Leveraging on similarity of packet structures and flow structures of frequently exchanged C&C flows within a P2P botnet, our proposed system initially uses clustering of flows and then Jaccard similarity coefficient on sample sets derived from clusters for accurate detection of bots. Ours is a very effective and novel framework which can be used for proactive detection of P2P bots within a monitored network. We empirically validated our model on traces collected from three different P2P botnets namely Nugache, Waledac and P2P Zeus.**

***Index Terms*—Bot, botnet, clustering, peer-to-peer.**

## I. Introduction

A bot is a malicious piece of software used to compromise a host in the network so that it can be remote-controlled by its master. A botnet is a coordinated group of bots to perform different malicious activities in the Internet at the behest of the "botmaster" [1], [2]. A botnet operates under common Command &Control (C & C) servers through establishment of C&C channels. For this, botmaster has to define some C&C protocol, which is the most intrinsic part of botnet's C&C strategy.

Botnet operators moving away from traditional chat based protocol like IRC to commonly used communication protocols like 'HTTP' and 'peer-to-peer' have made any direct communication between the botnet and the operators increasingly obscure. P2P botnets follow Peer-to-Peer (P2P) technologically and thus its distributed C&C structure makes it very resilient against detection. Our investigation in to the C&C behavior of P2P botnet has inspired us to formulate the following assumptions about its traffic pattern: (i) P2P botnet establishes numerous smaller sessions. For this, it frequently keeps on changing its communication ports; (ii) a P2P bot needs to keep communicating in order to keep its malicious network running. Moreover, all bots within a striving P2P botnet, periodically exchanges neighbor lists or peer list with each other to maintain a coherent network. A P2P botnet has following common botnet traits: (i) P2P bots like other bots follow a strict command-response pattern of communication, i.e. data flow occurs in both directions; (ii) Every botnet has its own specific set of commands and C&C interactions with the bots are preprogrammed to the set of commands they receive. Some of our assumptions are similar to the one used in Ref. [3], [4].

Our proposed system CluSiBotHealer is based on three important traits of P2P botnet's C&C traffic, namely frequency, repeatability and similarity. The system uses those payload independent statistical features of botnet's C&C traffic that exactly matches during an epoch for communicating P2P bots. Our system is solely based on detection of C&C channels through identification of corresponding network flows. A botnet's C&C channel is its weakest link and disruption of C&C channels will leave the botnet ineffective. A network flow provides essential information in a network like who is talking to whom i.e. conversation between hosts in the network. We define a flow by a combination of 5-tuple <source IP address, destination IP address, transport layer protocol, source port, destination port>. There are three phases in CluSiBotHealer: 1) Flow clustering phase - in which we cluster network flows using selected attribute values; 2) Flow reduction phase – in which we remove the duplicate flows from subject clusters and 3) Similarity analysis phase – in which similarity analysis between sets of flows derived in the previous step is done using Jaccard similarity coefficient.

We validated our model on C&C traffic collected from three prominent P2P botnets existing in the wild. Here is a small detail of the P2P botnets from which traffic samples were collected for this work. Nugache [5] is the pure-P2P bot artifact that does not depend on any central server including DNS. It handles C&C through encrypted P2P Channel using a variable bit length RSA key exchange, which is used to seed symmetric Rijndael-256 session keys for each peer connection. A new Nugache peer joins the network through an already known active servant peer in the network and each Nugache peer may maintain a list of up to 100 servant peers for future use in rejoining the network. Waledac [6] uses HTTP communication and a fast-flux based DNS network exclusively for its C&C operations. In order to make initial contact with the botnet, each Waledac binary carries a list of IP addresses to use as a bootstrap list. Additional resiliency is provided in Waledac binaries through a hardcoded URL to access the botnet in the event a bot is unable to find an active node in the bootstrap list. The domain used for the URL is part of the fast flux network created by the botnet. Each Waledac bot generates an internal public certificate and sends it through the botnet until it reaches the head-end C&C server.

Manuscript received July 21, 2014; revised December 20, 2014.

Pijush Barthakur is with the Department of Computer Applications, Sikkim Manipal Institute of Technology, Sikkim, India (e-mail: pijush.barthakur@gmail.com).

Manoj Dahal is with the Novell IDC, Bagmane Tech Park, C V Ramannagar, Bangalore, India (e-mail: mdahal@novell.com).

Mrinal Kanti Ghose is with the Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Sikkim, India (e-mail: mkghose2000@yahoo.com).

DOI: 10.7763/JACN.2015.V3.141

The C&C server uses the certificate to encrypt the current communication key required to interact with the botnet. Then the head-end C&C server sends the encrypted key back to the node. The Waledac node then decrypts this key and uses it for future communication with other nodes in the botnet. P2P Zeus or GameOver [7] is a P2P variant of its earlier popular centralized versions. The main P2P network is divided into several virtual sub-botnets by a hardcoded sub-botnet identifier in each P2P Zeus's bot binary. These sub-botnets are independently controlled by several botmasters, even though the main P2P network of Zeus is maintained and updated as a single entity. To make initial contact with the botnet, the bot binary carries a hardcoded list comprising of IP addresses, ports and unique identifiers of up to 50 Zeus bots. Peer list updating is done through a push-/pull- based peer list exchange mechanism. Zeus bot checks responsiveness of their neighbors every 30 minutes. Each neighbor is contacted in turn and given 5 opportunities to reply. If a neighbor does not reply within 5 retries, it is deemed unresponsive and is removed from the peer list. In case its entire neighbor becomes unresponsive, a Zeus bot attempts to re-bootstrap on to the network by contacting peers in its hardcoded peer list. If this also fails, the bot uses a DGA backup channel to retrieve a fresh RSA-2048 signed peer list.

The proposed botnet detection approach has following advantages: (1) CluSiBotHealer does not inspect packet payloads, which makes it free from privacy issues and also makes it work well with encrypted communication channels. (2) Unlike many other anomaly based approaches, our approach does not have to wait for specific anomalies to occur and hence can be effectively used for proactive detection of botnets. The kind of anomalies we are considering is inherent in the structure of botnet C&C flows and hence is available throughout a botnet's life cycle.

Rest of the paper is organized as follows: Section II provides a brief overview of related works. In Section III, we discuss the problem statement and system overview of CluSiBotHealer. This includes basic architectural overview, source and basic structural composition of data and an overview of basic features selected. Section IV describes the system detail which includes the process of dataset preparation, description of techniques used and the methodology of the proposed system. In Section V, the result of CluSiBotHealer has been discussed in detail. In Section VI we provide the conclusion of our work.

## II. Related Works

Despite having a significant increase in research on botnets in recent years, very few results have been adopted and implemented in real network scenarios [8]. Among the current botnet detection systems implemented for real network environment we have many well-known signature-based techniques [9]-[11]. Signature based detection leads to accurate detection of bots through comparison of every byte in the packet with that of known signature database. However, signature based detection can only detect known botnets. More importantly, signature based detection system may miss similar bots with slightly different signature. Another pioneering research group in the field of botnet that implemented in real network scenarios is the Honeynet project [12]. However, honeynets are found to be mostly useful in understanding botnet technology and characteristics, but do not necessarily detect bot infection [13], [14].

Many researchers have proposed botnet detection techniques using anomalies [15]-[17] that show up in the network because of botnet infection. In anomaly based detection approaches, the main idea is to detect botnets based on various anomalies observed in network traffic, such as, high traffic volume, high network latency, traffic on unusual ports, unusual system behavior etc. However, botnet detection solely based on anomalies may not be useful always for several reasons. First, anomalies may not always be prominent to indicate a botnet attack, particularly during early phase of infection. Second, it requires continuous monitoring of the network.

Problems faced in traditional ways of botnet detection, has motivated many researcher to try with automated and more reliable approaches. In Ref. [18], a data mining based framework called BotMiner detects bots through cross cluster correlation of similar communication pattern termed as C-plane and similar malicious activity patterns termed as A-plane. But, unlike A-Plane in BotMiner which involves noisy activities of bot in the network, CluSiBotHealer strives on detection of bots in its most silent state. Another data mining based approach, in Ref. [3], relies on application of few selected machine learning algorithms for detection of P2P bots. The result obtained is based on training of these algorithms using three hypotheses. CluSiBotHealer is a purely clustering based approach and using it we achieve far better accuracy. Ref. [19] proposed a machine learning based botnet detection approach using flow characteristics of IRC botnets. CluSiBotHealer uses packet and flow characteristics of P2P botnets and uses clustering unlike supervised methods used in [19]. In a recent work, conversation based P2P botnet detection "PeerShark" [20] has been proposed. PeerShark is a Port oblivious and Protocol oblivious technique that uses supervised learning algorithms. On the other hand CluSiBotHealer is highly dependent on identification of flows [5-tuple similarity] because in our view for identification of command-response pattern of communicating bots, knowledge of ports used by them is essential.

## III. Problem Statement and System Overview

### A. Basic Architectural Overview

A botnet's life cycle is clearly divided in to two phases, C&C phase and attack phase. At the very beginning, bot starts with bootstrapping to make initial contacts in the P2P network. Then it has to undergo "rallying", a mechanism through which bot identifies itself to the botnet's internal network. This establishes the bot's C&C channel with the botnet's server. Then the bot has to take several measures to secure it in the newly acquired host. For example, use of rootkit, anti-antivirus modules etc. A proactive detection approach has to deal with detecting a botnet during these early stages of infection. By proactive detection, we don't mean detecting a bot at the very start of infection. But, since a bot involves a lot of C&C interactions before it is put into malicious activities, our objective is to analyze bot's C&C

interactions passively so that it can possibly be detected before it participates in any form of attack. Therefore, we considered C&C flows as core of our design goal. C&C flows are present in a botnet throughout its lifecycle, starting from very beginning.

Every botnet uses a specific set of commands. Commands frequently exchanged between different peer bots represent flows whose structural characteristic matches with one another. These flows, when considered separately are low in volume, as very less number of packets is transferred and packet sizes are usually small. But, they are high in frequency, when we consider these flows together during an epoch. Therefore, we use clustering of flows using these structural characteristics in our flow clustering module. The architectural diagram of CluSiBotHealer is shown in Fig. 1. Then we use two additional modules Flow Reduction Module and Similarity Analysis Module for final detection of bots. In the flow reduction module, flows having same structural characteristics are removed. This enables us to assess the amount of reduction, which is usually very high in case of bots and also we get a 'set' of flows. In the similarity analysis module, we use Jaccard similarity coefficient to analyze similarity between such sets derived from probable bots.

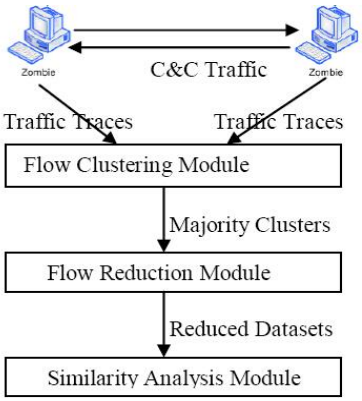



Fig. 1. Basic architectural model of CluSiBotHealer.

### B. Data Overview

The benign traffic samples were collected randomly from windows machines using Wireshark [21]. Our benign traffic samples include varied traffic such as HTTP, FTP, SMTP etc. We also include traffic captured from legitimate P2P application in our benign dataset. P2P file sharing involves rich web page transfers and normally carries packet to the size of MTU (*Maximum Transmission Unit*).

Botnet C&C traffic samples were collected from the following sources: The Nugache botnet C&C traffic was obtained from Department of Computer Science, The University of Texas at Dallas. This is the same botnet traffic sample used in the botnet related research works of [22]. Similarly, Waledac and P2P Zeus traffic traces were obtained from Department of Computer Science, University of Georgia. These traces were also used in the botnet related research works of [23]. A botnet's packet sizes are usually smaller and are seldom to the size of MTU.

### C. Feature Selection

We considered only high level features extracted from packet header. We define features (both packet level and flow level) based on network communication between hosts in the Internet, specifically in the context of botnets. Flows collected for an epoch (typically one day) are represented as $f_1, f_2, \ldots, f_m$ if $m$ flows are collected during the epoch $E$. Each $f_i$ is a collection of n packets sharing same TCP/UDP protocol, same source and destination IPs, same source and destination ports. Thus, $f_i = \{p_j\}_{j=1,\ldots,n}$ where each $p_j$ is single TCP/UDP packet. The flow level features are the aggregate of packet level features.

Our feature space consists of five features. These five features attain values that exactly match for the general and frequently exchanged set of commands during C&C interactions between peer bots. The statistical features involving time and interval of flows such as flow duration, starting time difference between two consecutive flows etc. are also important in the context of a botnet. But, we are not considering these features here because time and interval based features are dependent on many external factors like network bandwidth, congestion in the network etc. and may not exactly match for multiple bots in the network. Explained below are the features in our feature space:

1) Largest Size Packet or packet carrying maximum bytes in a flow (LSP): LSP is computed by comparing bytes transferred by packets in a flow, such that $p_i$ is any packet in that flow having highest bytes in its payload (where $i \leq n$ for $n$ packets in a flow).
2) Ratio of Largest Sized Packets in a flow (RLSP): If $m$ number of packets are carrying highest payload in a flow transferring in total $n$ packets, then RLSP is computed by dividing $m$ by $n$.
3) Average Packet Length (APL): APL is computed by dividing sum total of bytes transferred by all packets in a flow by total number of packets transferred in that flow.
4) Variance of Packet Length (VPL): If there are $n$ packets transferred in a flow, then variance calculated on number of bytes in payload of these packets is the value for VPL.
5) Response Packet Difference (RPD): We consider a pair of flows $f_i$ and $f_j$ as responding if the pair of source IP and source port in $f_i$ are destinations in $f_j$ and vice versa, while transport protocol remaining the same. If $n$ and $m$ are the number of packets transferred with $f_i$ and $f_j$ respectively, then RPD is the absolute difference between $n$ and $m$ i.e. $|n-m|$. We find many non-responding benign flows (one directional flow), for which we set a special value (e.g. a high value 999) to this feature.

## IV. System Details

### A. Dataset Preparation

We discarded certain categories of flows which are unlikely to contribute significantly in the process of clustering viz. (i) Flows having single packet, as it does not carry any meaningful information and (ii) flows that involves local broadcast activities in the network.

We prepared six datasets for three P2P botnets as follows: (i) we prepared one dataset of Nugache flows for each of the two Nugache bots. 15000 flows of each Nugache bot's C&C traffic were scaled between 0 and 1 for all the five features. Same has been done for 5000 flows of two separate sets of benign flows. Then these were labeled and combined so that each dataset has 20000 flows. (ii) We also prepared two

datasets each for Waledac and Zeus using the same procedure as described in (i). Waledac and Zeus datasets also has 15000 flows each and 5000 separate benign flows has been combined with each of these datasets. Furthermore, they are also labeled and scaled in the same way.

### B. EM Clustering Algorithm

Expectation-maximization (EM) clustering algorithm [24] is an iterative statistical method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on missing values. EM finds clusters by identifying a mixture of Gaussians that fit a given data set. The prior probability for each Gaussian is the fraction of points in the cluster defined by that Gaussian. These parameters can be initialized by randomly selecting means of the Gaussians, or by using the output of K-means for initial centers. The algorithm converges on a locally optimal solution by iteratively updating values for means and variances of Gaussians.

EM algorithm has two steps, defined as the expectation step (E-step) and maximization step (M-step). The missing labels are dealt with by alternating between the two steps. The expectation step involves fixing of models and estimation of missing labels. On the other hand, maximization step involves fixing of missing labels (or a distribution over the missing labels) and finding the model that maximizes the expected log-likelihood of the data.

### C. Jaccard Similarity Coefficient

Jaccard Index of similarity or Jaccard similarity coefficient is a statistical method for comparing the similarity of finite sample sets. It is calculated by dividing the size of intersections of sample sets by the size of its unions and is shown below:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},\ 0 \leq J(A, B) \leq 1. \quad (1)$$

### D. Methodology

We describe the methodology of our proposed system in the following steps:

Step 1: Network packets from two or more suspected machines are collected for same epoch (Typically one day). An epoch should be sufficiently long during day time when network usage is at its peak, so that it leads to accumulation of sufficiently large number of flows.

Step 2: Packets are grouped in to flows and preprocessed. We choose the features that can provide structural similarity of packets and flows. Our objective is to match same commands issued by the bots within the same botnet even though flows may be different because of frequent change of ports by the bots. Thus two or more datasets are prepared based on number of hosts under scanner.

Step 3: Expectation Maximization (EM) clustering algorithm is used to cluster the network flows of each dataset. Number of cluster to be generated is fixed at 'two' for obvious reason.

Step 4: If the difference in number of clustered instances among the two clusters is very high, it raises our initial suspicion that the host in question is a bot and the majority of the clustered instances in the larger cluster are bot flows. For example if more than 70% of the flows are clustered in to one cluster. In this case we tag the larger cluster as subject cluster for further evaluation.

Step 5: From each of the subject clusters we remove flows with duplicate feature values. If we get a significant reduction of flow instances, we tag the cluster and its corresponding host as a highly probable case of being a P2P bot. This is because large number of P2P bot flows shares the same packet and flow structure because of repeated transmission of same commands through different ports. Here, we need to discuss bot like benign traffic that might accidently be generated by some applications. Although, such flows might look similar, but it cannot exactly match even for the same applications running in two different hosts, because application running time is most likely to be different. Thus it will result in transfer of different number of packets, which in turn will result in different value for ratio of largest sized packets in a flow. However, this will not be the case for bot flows, because the number of packets having frequently exchanged bot commands in its payload is fixed, which means that for the number of times the bot gives the same command, the corresponding flows feature values will exactly match.

Step 6: Now that our subject clusters are left with only unique flow instances, we calculate the Jaccard similarity coefficient between pairs of subject clusters. If we get Jaccard index value greater than or equal to 0.1, we finally tag the host in the monitored network as bot.

## V. Results and Analysis

This section is divided in to two parts. We use WEKA [25] data mining environment in the first part. Weka provides a collection of Machine Learning (ML) algorithms and several visualization tools for data analysis and predictive modeling. In the second part we provide the final results of our detection model CluSiBotHealer. The results obtained in the first part are essential to validate the second part.

### A. Classes to Cluster Evaluation of Known C&C Flows

Here, we use Classes to clusters evaluation mode from Weka Explorer. In this mode Weka first ignores the class attribute and generate the clustering. Then during the test phase it assigns classes to the clusters based on majority value of the class attribute within each cluster. Then it computes the classification error, based on this assignment and also shows the corresponding confusion matrix. We use EM clustering algorithm to generate the clusters.

Results obtained are presented in Table1 using following performance metrics:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{Positive Predictive Value (PPV)} = \frac{TP}{TP+FP} \quad (4)$$

where *TP* = True Positives or Hits, *TN* = True Negatives or

correct rejections, *FP* = False Positives or false alarms and *FN* = False Negatives or misses.

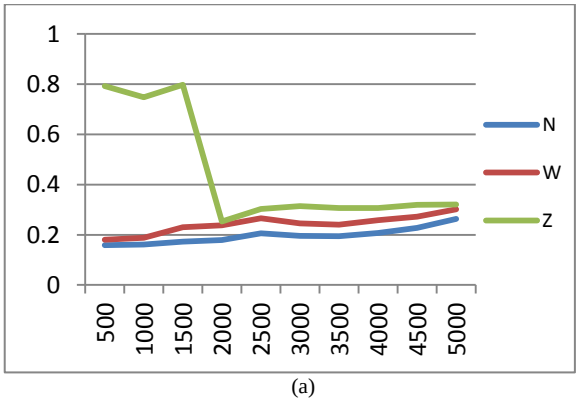
Here Sensitivity or Recall is the proportion of correctly identified bot flows out of total flows labeled as bot in our datasets. Similarly, PPV or Precision is the proportion of correctly identified bot flows out of total flows classified as bot by our classifier.

TABLE I: OUTPUT OF PERFORMANCE METRICS

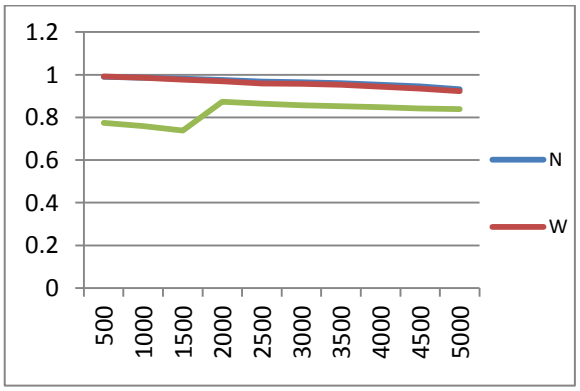
| | Nugache | Waledac | Zeus |
|---|---|---|---|
| Accuracy | 0.9321 | 0.922 | 0.83095 |
| Sensitivity | 0.997 | 0.99676 | 0.88163 |
| PPV | 0.919 | 0.90819 | 0.89172 |

Results in Table I shows that we achieved meaningful models from our labeled datasets. Therefore, in the next section we move on to propose a detection model through clustering of network flows which are not labeled previously and can be affectively used to judge a machine in a monitored network.

When we capture network flows from different subject machines, the benign traffic captured would be in different proportion to the bot C&C traffic. Availability of benign traffic depends on type of applications running in the monitored host. We also carried out an analysis using false positive rate and accuracy for different bot/benign traffic ratio. Highest number of benign flow in our dataset is 5000 in the ratio 1:3 to the number of bot flows. This we consider to be the base line considering the high frequency of bot C&C flows. Fig. 2 (a) and (b) represent change in false positive rate and accuracy respectively for different amount of benign flows.




(a)




(b)

Fig. 2. (a) Change in false positive rate for different amount of benign flows, (b) Change in accuracy for different amount of benign flows. Here, *N* is for Nugache, *W* for Waledac and *Z* for Zeus.

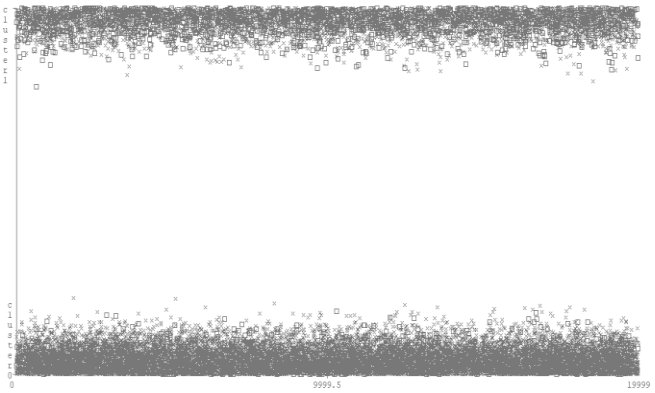


Fig. 3. Clusters generated for P2P Zeus. Cluster 0 indicate bot flows and cluster 1 indicate benign flows.

Though the results would vary depending on composition of benign dataset, we achieved best results for the ratio 1:7 between benign and bot flows.

In Fig. 3, X-axis represent number of instances and Y-axis represent clusters. Squares in the figure indicate wrongly clustered instances.

### B. *Analysis of Results Obtained from CluSiBotHealer*

In this section, we apply our botnet detection framework CluSiBotHealer on datasets prepared from flows features of our captured traffic. The system processes through three consecutive modules for final detection of bots. Described below are the results of the three modules:

*Clustering of C&C flows (Module 1):* Flows collected from subject machines in the monitored network are clustered using EM clustering algorithm. For proper evaluation and demonstration we used a specific number of flows in each case. The EM clustering algorithm is configured (in our case) to generate two clusters. We analyzed majority clusters (the bigger cluster) generated from subject machines. Table II shows the number of flows in percentage in majority clusters in each case.

Our main interest is in majority clusters because it is likely to hold botnet C&C flows. We also observe that bot infected machines generates highly imbalanced clusters when compared with clusters generated from flows that belong to benign machines. Therefore, in our next module we only consider the majority clusters for further analysis.

TABLE II: PERCENTAGE OF FLOWS IN MAJORITY CLUSTERS

| | |
|---|---|
| Nugache Bot1 | 81.385 |
| Nugache Bot2 | 81.395 |
| Waledac Bot1 | 82.385 |
| Waledac Bot2 | 82.245 |
| Zeus Bot1 | 74.845 |
| Zeus Bot2 | 73.455 |
| Benign Machine1 | 65.28 |
| Benign Machine 2 | 59.88 |

*Removal of duplicate flows (Module 2):* From majority clusters we removed duplicate flows to create sample sets of flow instances. Duplicate removal is based on exact matching of values for five features described in Section III. Table III shows the percentage of reduction achieved in each case.

Majority clusters derived from bot shows huge reduction in its volume because of repetitive C&C messages that go around it very frequently. This is because a P2P bot has to open up for communication with all its peers in the same way.

Some benign clusters may also show significant reduction in size depending on application running on it at the time of traffic capture. Therefore, we find a significant difference in reduction rates of Benign Machine 1 and Benign Machine 2 in Table III. These reduced sample sets are used in our next module to compute Jaccard similarity coefficients.

TABLE III: PERCENTAGE OF REDUCTION IN EACH MAJORITY CLUSTER AFTER DUPLICATES ARE REMOVED

| | |
|---|---|
| Nugache Bot1 | 96.6 |
| Nugache Bot2 | 96.78 |
| Waledac Bot1 | 92.85 |
| Waledac Bot2 | 91.88 |
| Zeus Bot1 | 88.25 |
| Zeus Bot2 | 86.71 |
| Benign Machine1 | 26.1 |
| Benign Machine 2 | 82.96 |

*Determination of cluster similarity (Module 3):* We calculated Jaccard similarity coefficient between sample sets derived from majority clusters. We get significantly higher similarity between bots that are part of same botnet. Table IV shows Jaccard similarity coefficient between bots. The Jaccard similarity coefficient between Benign Machine 1 and Benign Machine 2 is 0.0195 and is significantly lower than bot similarity as shown in Table IV.

TABLE IV: JACCARD SIMILARITY COEFFICIENT BETWEEN BOTS

| | Nugache Bot2 | Waledac Bot2 | Zeus Bot2 |
|---|---|---|---|
| Nugache Bot1 | 0.1926 | 0.0197 | 0.015 |
| Waledac Bot1 | 0.0231 | 0.2157 | 0.011 |
| Zeus Bot1 | 0.0155 | 0.0098 | 0.1008 |

Therefore, our heuristically proposed baseline for botnet detection using CluSiBotHealer is: Majority cluster having ≥ 70 % of total flows, flow reduction ≥ 80 % and Jaccard similarity coefficient of sample sets derived from majority cluster is ≥ 0.1. If all these three conditions are satisfied for flows collected from a suspected host, we consider the host to be bot infected.

## VI. CONCLUSION AND FUTURE WORK

A botnet detection model called "CluSiBotHealer" has been proposed. CluSiBotHealer is based on three important traits of P2P botnet's C&C architecture: i) When C&C flows are clustered from bot infected machines we get highly imbalanced clusters, ii) when duplicate flows are removed from majority clusters of bot infected machines, we get very high reduction in its volume, and iii) we get high Jaccard similarity coefficient for sample sets derived from majority clusters for bots of same botnet. If all these three conditions are satisfied in sequence, we can conclude that the machine from which flows were clustered is bot within monitored network. Heuristic values for the three conditions are as follows: Majority cluster having ≥ 70 % of total flows, flow reduction in majority cluster ≥ 80 % and Jaccard similarity coefficient among sample sets derived after flow reduction in majority cluster is ≥ 0.1.

Furthermore, we also carried out a clustering analysis of known botnet C&C flows. We achieved good accuracy, sensitivity and PPV on known samples. This inspired us to develop CluSiBotHealer for detection of unknown botnets in the wild. However, we need to carry out more experiments in large network environment. In our view, CluSiBotHealer can be used in specially designed honeypots to capture the traffic properties of unknown botnet C&C flows. Moreover, we can use these flows to derive additional properties which can then be used to generate efficient machine learning based classification models.

## ACKNOWLEDGMENT

We would like to thank Mohammad M. Masud, Department of Computer Science, University of Texas at Dallas for providing us a botnet traffic sample to carry out this research work. We also thank Babak Rahbarinia, Department of Computer Science, University of Georgia, USA, for giving us another two samples of P2P botnet traffic.

## REFERENCES

[1] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: understanding, detecting, and disrupting botnets," *SRUTI*, 2005.
[2] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnetphenomenon," *ACM IMC*, 2006.
[3] W. H. Liao and C. C. Chang, "Peer to peer botnet detection using data mining scheme," in *Proc. International Conference on Internet Technology and Applications*, 2010, pp. 1-4.
[4] G. F. Gu, V. Yegneswaran, P. Porras, J. Stoll, and W. Lee, "Active botnet probing to identify obscure command and control channels," in *Proc. Annual Computer Security Applications Conference*, 2009.
[5] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich, "Analysis of the storm and nugache trojans: P2P is here," *USENIX*, vol. 32, no. 6, December 2007.
[6] G. Sinclair, C. Nunnery, B. Byung, and H. Kang, "The waledac protocol: the how and why," in *Proc. 4th International Conference on Malicious and Unwanted Software*, Feb. 2010.
[7] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, "Highly resilient peer-to-peer botnets are here: an analysis of gameover zeus," in *Proc. the 8th IEEE International Conference on Malicious and Unwanted Software*, October 2013.
[8] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Computer Networks*, vol. 1, 2012.
[9] K. Rieck, G. Schwenk, T. Limmer, T. Holz, and P. Laskov, *Botzilla: Detecting the Phoning Home of Malicious Software*, SAC, 2010.
[10] J. Goebel and T. Holz, "Rishi: identify bot contaminated hosts by IRC nickname evaluation," in *Proc. the First Conference on First Workshop on Hot Topics in Understanding Botnets*, USA, 2007, p. 8.
[11] Snort-Snort 2006. [Online]. Available: http://www.snort.org
[12] Project and research alliance. Know your enemy: Tracking Botnets. (March 2005). [Online]. Available: http://www. honeynet.org/papers/bots/.
[13] M. Feily, A. Shahrestani, and S. Ramadass, "A Survey of botnet and botnet detection," in *Proc. Third International Conference on Emerging Security Information*, Systems and Technologies, 2009.
[14] H. R. Zeidanloo, M. J. Zadehshooshtari, P. V. Amoli, M. Safari, M. Zamani, "A taxonomy of botnet detection techniques," in *Proc. ICCSIT 3rd IEEE International Conference*, 2010.
[15] J. R. Binkley and S. Singh, "An algorithm for anomaly-based botnet detection," in *Proc. the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet*, 2006, vol. 2, p. 7.
[16] Y. Yang, G. Y. Hu, and S. Z. Guo, "Imbalanced classification algorithm in botnet detection" in *Proc. First International Conference on Pervasive Computing, Signal Processing and Applications*, 2010.
[17] P. Burghouwt, M. Spruit, and H. Sips, "Detection of botnet collusion by degree distribution of domains," in *Proc. ICITST*, 2010.
[18] G. F. Gu, R. Perdisci, J. J. Zhang, and W. K. Lee. "Botminer: clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. 17th USENIX Security Symposium*, 2008.
[19] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using machine learning techniques to identify botnet traffic," in *Proc. 2nd IEEE LCN Workshop on Network Security*, Nov. 2006.
[20] P. Narang, S. Ray, and C. Hota, "Peershark: detecting peer-to-peer botnets by tracking conversations," in *Proc. IEEE Security and Privacy Workshops*, 2014.
[21] Wireshark homepage. [Online]. Available: http://www.wireshark.org

[22] M. M. Masud, J. Gao, L. Khan, J. W. Han, and B. Thuraisingham, "A multi-partition multi-chunk ensemble technique to classify concept-drifting data streams," in *Proc. the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2009.
[23] B. Rahbarinia, R. Perdisci, A. Lanzi, and K. Li, "PeerRush: mining for unwanted p2p traffic," in *Proc. 10th Conference on Detection of Intrusions and Malware & Vulnerability Assessment*, 2013.
[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 39, no. 1, pp. 1-38, 1977.
[25] Waikato homepage. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/

**Pijush Barthakur** received the master of computer application (MCA) degree from Dibrugarh University, India in 2001. Currently, he is working as an associate professor at Department of Computer Applications, Sikkim Manipal Institute of Technology, Sikkim, India. He is also pursuing his doctoral degree in Sikkim Manipal University.

His research interests are in the area of network security. Two of his earlier publications are also in the same area. "A framework for P2P botnet detection using SVM," in the 4th international conference on cyber-enabled distributed computing and knowledge discovery (CyberC), 2012 and "An efficient machine learning based classification scheme for detecting distributed command & control traffic of P2P botnets," IJMECS, vol. 5, no. 10, pp. 9-18, 2013.

Mr. Barthakur had also been a member of Technical Program Committee at 5th International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, China, 2013.

**Manoj Dahal** received his Ph.D. degree in networking from Tezpur University, India in 2008 for the thesis on addressing transport layer congestion control issues. Currently, he is working at Novell, India and his professional work mostly lays on file access protocol areas.

He is also associated with research on detection of botnets using machine learning techniques at Sikkim Manipal Institute of Technology, Sikkim, India. He has around 15 years of experience in software industry. He was a post-doctoral fellow for about a year with Inria, France at LIP Labs, ENS de Lyon, where he has worked on traffic engineering for optical networks. He also worked as a professor for a short period in the Department of Computer Science and Engineering, at Sikkim Manipal Institute of Technology, Sikkim. He also worked with Nokia (via Satyam) on routing devices and National Informatics Centre on e-Governance Projects in India.

**M. K. Ghose** is currently the dean (Academics) of SMIT and professor in Department of Computer Science & Engineering at Sikkim Manipal Institute of Technology, Majitar, Sikkim, India. Formerly he was the dean (R&D) since June, 2006. From June 2008 to June 2010, he had also carried out additional responsibilities of Head, SMU-IT.

Prior to this, he worked in the internationally reputed R & D organization ISRO – from 1981 to 1994 at Vikram Sarabhai Space Centre, ISRO, Trivandrum in the areas of mission simulation and quality & reliability analysis of ISRO launch vehicles and satellite systems and from 1995 to 2006 at Regional Remote Sensing Service Centre, ISRO, IIT Campus, Kharagpur in the areas of RS & GIS techniques for the natural resources management. He was also associated with Regional Engg. College (NIT), Silchar (1979 – 1981) as teaching Asst. and Assam Central University, Silchar as COE and HOD of Computer Science Department (1997 - 2000). His areas of research interest are data mining, simulation & modeling, network, sensor network, information security, optimization & genetic algorithm, digital image processing, remote sensing & GIS and software engineering. He chaired a number of national/international conference sessions. He has conducted quite a number of seminars, workshop and training programs in the above areas and published 126 technical papers in various national and international journals in addition to presentation/ publication in several international/ national conferences. Till date, he has produced 8 Ph.Ds. and research assistance given for 2 Ph.Ds. Presently 11 scholars are pursuing Ph.D. work under his guidance.

Dr. Ghose is having 8 sponsored projects worth of 1 crore (INR). Dr Ghose also served as a technical consultant to various reputed organizations like IIT Chennai, IIT Kharagpur, WRI, Tricy, SCIMST, KELTRON, HLL, Trivandrum.