SCIS 2014 The 31th Symposium on
Cryptography and Information Security
Kagoshima, Japan, Jan. 21-24, 2014
The Institute of Electronics,
Information and Communication Engineers

Semi-supervised Botnet Detection Using Ant Colony Clustering

Khalid Huseynov* Kwangjo Kim* Paul D. Yoo†

Abstract: Recently, botnets have become one of the fast growing and changing vectors of malicious underground economy. They pose serious threats on the cyber-security of citizens, enterprises, and governments. Many recent countermeasures utilize machine-learning techniques due to its adaptability and "model-free" properties. In this research, we propose a bio-inspired computing technique called ant colony clustering for the accurate, scalable detection of botnet attacks. The proposed method is able to detect the botnet hosts rapidly and accurately while not depending on its traffic payload. Furthermore, it utilizes only a small sample of labeled data in the form of semi-supervised learning.

Keywords: botnets, intrusion detection, ants-based clustering, P2P, swarm intelligence

1 Introduction

A botnet represents a group of compromised host machines called bots which are controlled remotely by an originating botmaster server. The botmaster is not only able to download any confidential files but also capable to execute any malicious code on the infected machines, which turns the botnet into a platform of massively coordinated cyber-attacks. Bots can perform any kinds of malicious attacks such as Distributed Denial of Service (DDOS), click-fraud, adware, spreading spam, key logging, and stealing personal information. According to report by Damballa, a cyber-security company, few millions of computers in the United States were infected by botnets in 2009 (*e.g.,* 3.6M by *Zeus* botnet) [1].

The two key components of botnets are the protocols employed for communication and architecture. The botnets in early 2000's used Internet Relay Chat (IRC) as a communication protocol, which refers to centralized architecture. In this scenario, botmaster is able to communicate with its bots in real time via chat, through IRC-based Command and Control (C&C) server. In the middle of 2000's, HTTP-based centralized botnet architectures [2, 3, 4] emerged. In such architectures, bots periodically contact their C&C server to receive further instructions using HTTP protocol as a basis. The main drawback of centralized architecture is the ease of take down by mere shutdown of C&C server once detected.

Alternatively, botnets were also using Peer to Peer (P2P) architecture and protocols that evolved in the mid-2000s. In this scheme, the commands of botmaster can pass through multiple bots in order to reach their destinations, and if some of the bots are down, another path is selected based on its P2P protocol. *Kademlia* is a good example of such P2P protocols [5]. Nugache, Storm and Waledac [6, 7] are also well-known P2P botnets.

Various defending mechanisms against botnets have been developed and are being utilized these days. Although signature-based methods have gained popularity [8] for certain botnets' detection, current antivirus software is unable to detect a few subtle behaviors between bots. Furthermore, bots are updated periodically by its botmaster so that the signatures collected by security companies become outdated. Therefore, with botnets, the signature-based detection mechanisms are not effective. Another well-known method of botnet detection could be the botnet infiltration [9]. This

* Computer Science Dept., KAIST, 291 Gwahak-ro, Yuseong-gu, Daejeon, 305-701, Korea. {khalid.huseynov, kkj}@kaist.ac.kr).
† Dept. of ECE, Khalifa Univ., PO Box 127788, Abu Dhabi, UAE. paul.d.yoo@ieee.org

reverse engineering approach includes sampling malware code and understanding the communication syntax of a particular botnet. However, this approach is not shown to be scalable since the whole process should be repeated for every new type of botnet. In addition, the complexity of such reverse engineering approach increases as the malware becomes complex. The other popular approach to botnet detection includes the behavior analysis of network traffic [10, 11, 12]. Such methods are focused on learning the patterns of network flows using data-mining and machine-learning techniques. Thus, clustering, classification, or correlation methodologies can be applied to the similarity behavior measure of bots in the form of anomaly detection.

Recent advances in swarm intelligence (SI) – a bio-inspired family of techniques – allowed us to take a close look into the SI-based data-mining methods. The most promising algorithms developed in such technique are based on the behavior of ants as a colony [13]. A few successful applications of ant colony clustering (ACC) were reported in the literature, namely, data retrieval and textual document clustering [14], web usage mining [15, 16], network traffic analysis [17], intrusion detection [18] and biomedical data processing [19]. To the best of our knowledge, there was no application of ant-based algorithms in the botnet detection.

Our detection technique is built on the fact that the bots within the same botnet behave similarly in terms of network traffic behavior. Using ACC-based unsupervised-learning algorithm, we found the feature clusters of botnet traffic. We have also successfully identified the clusters using a sample of labeled data from original dataset. Combining such two methods could be seen as a "*semi-supervised approach*". Again, our method does not depend on the payload, and therefore is capable to detect even botnets utilizing encrypted communications. Since we divide the communication between hosts (flows) into intervals, the detection can be made rapidly. Such method is also applicable to centralized (IRC and HTTP) as well as P2P botnets. Note that our method does not detect the bots at the infection stage when botnet is set up and communicating with C&C server.

2 Related Work

As discussed, most detection methods in the literature are either signature, machine-learning, or reverse engineering-based. Rishi [8] is a well-known signature-based botnet detector in IRC channels. It has been built on the concept that host machines after infected contact their C&C server with their nicknames. Here, nicknames are used for further identification of the bots in the botnet. Such nicknames usually contain some constant parts, which are the same for all the bots in the same botnet. Such simple idea was shown to be effective for IRC botnet detection in a network with its speed up to 10Gbit/sec.

A successful attempt to infiltrate a huge botnet system called Torpig was conducted in [9]. This work is of particular interest since they infiltrated Torpig C&C server and were able to record all the communications of its bots. They successfully identified 1.2 million IP addresses of bots, which are connected to an infiltrated C&C server.

A number of methods based on the analysis of network traffic have also been proposed. Gu *et al.* proposed BotHunter [10] that relies on botnet lifecycle activities, namely, scanning, infection, binary download, and C&C scanning. BotHunter utilizes a Snort-based intrusion detection system for any kind of scanning. Once successfully detects, it inspects the payload of flow for other malicious activities from botnet lifecycle. Encrypted packages are the obstacles for effective functioning of BotHunter.

Gu *et al.* enhanced the BotHuter and named it BotMiner [20], which was built on the assumption that all the bots in the botnet exhibit similar network behavior. This is similar to our assumption. BotMiner searches for and clusters the similar connections using a C-plane monitor. Similar activities are clustered using an A-plane monitor. BotMiner then cross-correlates the two planes and finds which C-plane cluster behaves maliciously. Note that traditional and simple clustering methods were used in BotMiner. BotMiner achieved the accuracy of 99% and its false positive rate was around 1%.

Wang *et al.* proposed another traffic analysis model for P2P botnet detection. They observed that botnet control flows are relatively more stable compared to normal flows. As a result, their algorithm was able to detect the bots using encrypted communication with high true positive rate and low false alarms.

A novel approach based on both traffic behavior analysis and flow intervals was proposed in [21]. The flows were organized based on time intervals. In addition, a machine-learning based classifier was built based on the extracted traffic flow features. This approach was evaluated on a labeled dataset

containing malicious traffic from Storm and Waledac P2P botnets, and showed 99% accuracy and less than 1% false alarm rate.

Our approach employed flow intervals as well. However, instead of building a supervised classifier, which requires large amount of labeled dataset, we utilized partially labeled dataset and promising ACC-based algorithm was applied.

3 Approach and Methodology

3.1 Overview

Our approach is based on the network packet analysis, opposite to signature-matching approaches. Our approach is built on the assumption that the there is a significant level of similarity in the network behavior of bots in the same botnet. The bots in the same botnet are running on the same malicious code and utilize the same protocol for communications. Therefore, the traces of communication with botmaster should be analogous to one another. In some cases, botmasters use its botnet as a platform for *ddos* (*e.g.*, syn flood) attacks, meaning that all the computers in the botnet behave very similarly. Thus, we can take advantage of such similarity from either in communication patterns or malicious activity patterns. Some recent approaches take advantage of both [20], resulting in unsupervised method. Most Intrusion Detection Systems (IDSs) take advantage of such similarity in activity patterns. However, our approach is looking at the communication pattern of bots. Our approach will be elaborated in the following section.

The first stage of our approach is the extraction of features representing communication patterns of bots. We then perform data cleaning, and finally apply the novel clustering technique to group similar activities together. During this stage, the traffic corresponding to botnets should be clustered. Figure1 displays the step-by-step procedure of our approach.

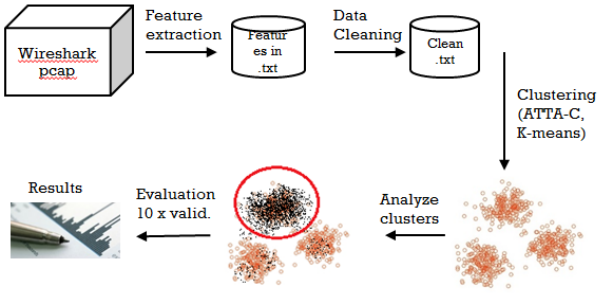

Figure 1. Overview of the approach

The crucial difference of our approach lies in the selection of clustering algorithm. Existing botnet detection methods [10, 11, 20] use popular data-mining techniques. However, our aim is to see the usability of ACC-based algorithm in botnet detection. To our best knowledge, the use of ACC-based algorithm in the application of botnet detection is novel. The details of the ACC algorithm selection and its theory will be given in Section 3.3.

Having the clusters identified, we need to label whether it is malicious or benign. If we have one single botnet, we may need to find one cluster only. However, if the traffic contains multiple types of botnets, a multiple malicious clusters may be found. For the identification of malicious clusters, we sampled 10% of the original labeled data. The dataset we selected for our evaluation [21] is labeled; therefore, we chose only 10% subset, and the remainder was left unlabeled, leading to semi-supervised approach. Our semi-supervised approach more resembles the real world situations where traffic labels are not always available. The dataset we used will be discussed extensively in Section 3.4.

3.2 Feature Selection

Finding right features is of great importance for any classification problems. The features used in our experiments are extracted from the flow intervals. Each flow is represented by source ip, destination ip, source port, destination port, and transport layer protocol (*e.g.,* tcp). These flows are then divided into the intervals of length, T. Thus, each extracted record corresponds to the above-mentioned 5-tuple data coupled with the interval T.

The features we selected are independent from traffic payload, and represent the communication patterns of botnet traffic. For example, it was observed that botnet communications result in many uniformly sized, small packets [21]. The communication in the initial stage of protocol (when the host gets infected) has also been observed to vary from behavior of the rest of the traffic. Table 1 gives the detailed information on 11 selected features.

Table 1. Selected features and descriptions

Attribute	Description
SrcPort	Flow source port address
DstPort	Flow destination port address
Proto	Transport layer protocol or mixed

APL	Average payload packet length for time interval
PV	Variance of payload packet length for time interval
PX	# of packets exchanged for time interval
PPS	# of packets exchanged per second in time interval T
FPS	The size of the first packet in the flow
TBP	The average time between packets in time interval
NR	# of reconnects for a flow
FPH	# of flows from this address over the total number of flows generated per hour

Note that the features were extracted from connection log generated by Bro IDS [22] using python.

3.3 Adaptive Time Dependent Transporter Ants Clustering (ATTA-C)

Ant-based clustering is inspired from the brood sorting behavior of ants. It was first developed by Deneubourg [13] for robotics modeling. The key concept is that all the data available first randomly are distributed on 2-D grid. N ants (agents) are then assigned the random coordinates on the grid. The ants can pick up and drop data items. The probability of picking an item is increased if a data is surrounded by dissimilar data. On the other hand, the probability of dropping an item is increased if ants are surrounded with similar data. The following formulas correspond to those probabilities.

$$P_p = \left(\frac{k_1}{k_1 + f}\right)^2 \qquad P_d = \left(\frac{f}{k_2 + f}\right)^2$$

where

- P_p and P_d are the probabilities of picking up and dropping data item, respectively;
- k_1 and k_2 are the threshold items;
- f is the perceived neighborhood density of an ant.

The major advancements were added by Lumer and Faieta (LF) [23]. They introduced the dissimilarity-based evaluation of the local density function f and the notion of short-term memory within each agent. Recent research improved it even further with ATTA-C model [24]. The ATTA-C is one of the few algorithms that have been benchmarked on various datasets, and is now publicly available under GNU agreement. However, none of the datasets ATTA-C has been tested relates to network traffic data. Therefore, our work sheds light on suitability of ATTA-C for network traffic clustering.

3.4 ISOT Dataset

ISOT dataset was created by Information Security and Object Technology (ISOT) research lab at the University of Victoria [21]. Basically, this is a mix of several existing open (malicious and non-malicious) datasets. The malicious traffic in ISOT dataset obtained from French chapter of honeynet project [25] and includes Storm and Waledac botnets. Storm botnet had its peak in 2007 - 2008 with more than a million infected bots. In addition, Waledac was considered as the successor of Storm with well distributed P2P style communication protocol. Unlike overnet used by Storm, Waledac utilizes HTTP communication and fast-flux DNS network.

Non-malicious traffic was collected from two sources. One was obtained from the Traffic Lab at Ericsson Research in Hungary [26] (everyday usage traffic). This traffic was integrated with second dataset, which built by Lawrence Berkeley National Lab (LBNL) [27]. This combination is important since Ericsson Lab dataset includes general traffic from a variety of applications as well as HTTP web browsing, World of Warcraft traffic, and traffic from Azureus bittorent client. On the other hand, LNBL traffic comes from a medium-sized enterprise network and consists of five large datasets.

In total, ISOT dataset contains 14.1 GB of Wireshark *pcap* format network trace. After feature extraction, it was reduced to 104.4 MB.

4 Results and Discussion

Since flows are based on time intervals, we extracted features for 5 seconds (s) intervals from 30s to 500s. Starting from 300s the accuracy flattens out. Figure 2 describes the dependence of accuracy from the length of flow interval in more details. All the results are presented with the time interval of T = 300s.

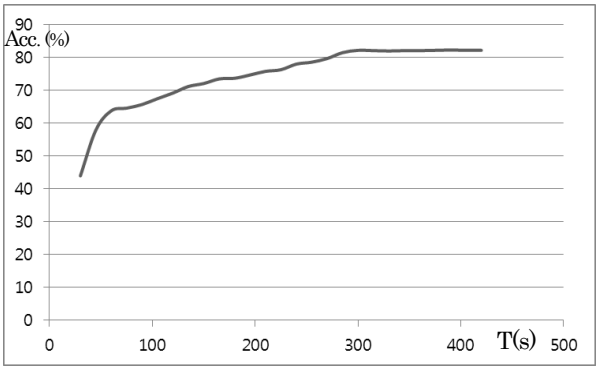

Figure 2. Dependence of accuracy from time interval of a flow.

Table 2 presents our preliminary results and their comparison with other detection methods.

Table 2. Comparison with other methods.

Method		Benchmark Dataset	True positive	False positive	Running time
Our approach	K-means	ISOT	82.1%	2.4 %	1~2
	ATTA-C	ISOT	67.8%	23.5%	274
D.Zhao et al.		ISOT	98.3%	0.01 %	N/A
Botminer		N/A	99.6%	0.3%	N/A

We have tested our approach using two different clustering algorithms, namely, ATTA-C and K-means for comparison.

As observed, ATTA-C does not scale well for larger datasets. The largest benchmark dataset used so far is presented in Dorigo *et al.*, which contains 3, 498 records ('Digits' dataset). However, our dataset (T = 300s) has around 681, 203 records, 7, 193 of which are malicious. Therefore, we were not able to run ATTA-C on this dataset directly. The results of ATTA-C are obtained from sampling 1% of the full dataset, 6, 700 records of normal and 80 records of malicious traffic. As depicted, the preliminary results for ATTA-C do not exhibit high detection accuracy. However, these results are distorted with the small sample size.

To test our hypothesis, we also applied K-means algorithm in the clustering stage. With K-means, we used the whole dataset. The true positives of 82.1% shows that the samples applied in ATTA-C do not represent the whole dataset correctly. In other words, there can be noise in original data. However, 82.1% does not represent high accuracy by itself. Therefore, we can conclude that some level of noise presents in the dataset and needs to be further cleaned.

Another problem of ATTA-C is the usage of agglomerative clustering at the last stage of clustering process. Agglomerative clustering itself is not a problem; however, the absence of control mechanism for threshold is a crucial drawback.

One of the possible reasons for such a slow performance of ATTA-C on large datasets could be its large memory usage. Note that for N records the grid where ants move is 10N x 10N, meaning 7, 000, 000 by 7, 000, 000. Moreover, we have to keep all the data records in main memory as well, meaning *thrashing* can occur.

5 Conclusion

In this paper, we presented semi-supervised botnet detection method and evaluation of the new type of bio-inspired algorithm in the context of network traffic analysis. Since there haven't been similar approaches in botnet detection, we aimed to measure the effectiveness of ATTA-C algorithm in botnet detection. As discussed, ATTA-C algorithm is not scalable with larger datasets, meaning current version cannot be used for online traffic analysis due to the high volumes of data. We discussed a few causes of the poor performance of ATTA-C such as memory thrashing, data noise, and absence of control mechanism for agglomerative clustering threshold.

Our approach was shown to be payload independent, meaning that it can detect the bots employing encrypted communication. Furthermore, the existence of botnet can be detected within a short amount of time, around 300s. This is much faster compared to the methods using flows only.

As our future work, we plan to improve current approach by finding solutions for the afore-mentioned problems. Since ACC-based algorithms have been applied in many different fields, we plan to modify existing algorithms for the needs of our field, such as model scalability and efficiency to deal with large volumes of data.

Currently, bio-inspired ant-based computing is an active field of research. Therefore, its application in the fields like intrusion/botnet detections may outperform currently existing methods. Moreover, ant-based algorithms can be modified to work in distributed and parallel manner, which is a crucial criterion for large-scale detection methods.

Acknowledgement

This research was supported by the KUSTAR-KAIST Institute, Korea, under the R&D program supervised by the KAIST and funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

References

[1] E. Messmer, "America's 10 most wanted botnets", Damballa, Atlanta, GA, 2009 http://www.networkworld.com/news/2009/072209-botnets.html, Accessed: November 2013

[2] K. Chiang and L. Lloyd, "A case study of the restock rootkit and spam bot", In *Proceedings of USENIX HotBots'07*, 2007.

[3] N. Daswani and M. Stoppelman, "The anatomy of clickbot.a", In *Proceedings of USENIX HotBots'07*, 2007.

[4] N. Ianelli and A. Hackworth, "Botnets as a vehicle for online crime." http://www.cert.org/archive/pdf/Botnets.pdf, Accessed: November 2013.

[5] P Maymounkov and D Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric", *Peer-to-Peer Systems*, 2002

[6] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, "Peer-to-peer botnets: Overview and case study", In *Proceedings of USENIX HotBots'07*, 2007.

[7] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling., "Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm", In *Proceedings of the First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'08)*, 2008.

[8] J. Goebel and T. Holz, "Rishi: Identify bot contaminated hosts by irc nickname evaluation", In *Proceedings of USENIX HotBots'07*, 2007.

[9] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski , R. Kemmerer , C. Kruegel , and G. Vigna, "Your Botnet is My Botnet: Analysis of a Botnet Takeover", In *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 635-647

[10] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: Detecting malware infection through ids-driven dialog correlation", In *Proceedings of the 16th USENIX Security Symposium (Security'07)*, 2007.

[11] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting botnet command and control channels in network traffic", In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, 2008.

[12] M. K. Reiter and T.-F. Yen, "Traffic aggregation for malware detection", In *Proceedings of the Fifth GI International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, (DIMVA'08)*, 2008.

[13] J.-L. Deneubourg, S. Gross, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien, "The dynamics of collective sorting: Robot-like ants and ant-like robots", *In Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, Cambridge, MA, MIT Press, 1991, pp. 356-363.

[14] V. Ramos and J.J. Merelo, "Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning", In AEB'2002, *First Spanish Conference on Evolutionary and Bio-inspired Algorithms*, Spain, 2002, pp. 284-293.

[15] A. Abraham and V. Ramos, "Web usage mining using artificial ant colony clustering and linear genetic programming", In *Proc. Congress on Evolutionary Computation (IEEE Press)*, Australia, 2003, pp. 1384-1391.

[16] H. Azzag, G. Venturini, A. Oliver, and C. Guinot, "A hierarchical ant based clustering algorithm and its use in three real-world applications", *European Journal of Operational Research*, Vol. 179, Issue 3, 16 June 2007, pp. 906-922.

[17] T. Ekola, M. Laurikkala, T. Lehto, and H. Koivisto, "Network Traffic Analysis Using Clustering Ants", *World Automation Congress* (WAC 2004), Sevilla, Spain, June 2004.

[18] V. Ramos and A. Abraham, "Antids: Self-Organized Ant-Based Clustering Model for Intrusion Detection System", In *Proceedings of the Fourth IEEE International Workshop*, WSTST'05, Muroran, Japan, 2005, pp. 977-986.

[19] M. Bursa and L. Lhotska, "Ant Colony Inspired Clustering in Biomedical Data Processing", In *3rd European Symposium on Nature-inspired Smart Information Systems* [CD-ROM], Aachen: NiSIS, 2007.

[20] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection", In *Proceedings of the 17th USENIX security symposium*, San Jose, CA, USA 2008.

[21] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, and D. Garan, "Botnet detection based on traffic behavior analysis and flow intervals", *Computers & Security* (2013).

[22] V. Paxson, "Bro: a system for detecting network intruders in real-time", *Computer networks*, 1999.

[23] E.D. Lumer and B. Faieta, "Diversity and adaptation in populations of clustering ants", Cambridge: MIT Press, In D. Cliff, P. Husbands, J.-A. Meyer, & S.W. Wilson (Eds.), From animals to animats: *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, 1994, pp. 501-508.

[24] J. Handl, J. Knowles, and M. Dorigo, "Ant-Based Clustering and Topographic Mapping", Artificial Life, MIT Press, Vol. 12, No. 1, 2006, pp.35-61.

[25] French Chapter of Honeynet: http://www.honeynet.org/chapters/france. Accessed: November, 2013

[26] G. Szab´o, D. Orincsay, S. Malomsoky, and I. Szab´o, "On the validation of traffic classification algorithms", In *Proceedings of the 9th international conference on Passive and active network measurement*, PAM'08, (Berlin, Heidelberg), pp. 72–81, Springer-Verlag, 2008.

[27] LBNL Enterprise Trace Repository. http://www.icir.org/enterprise-tracing. Accessed: November 2013