Unsupervised Hadoop-based P2P Botnet Detection with Threshold Setting

Khalid Huseynov* Kwangjo Kim*

Department of Computer Science, Korea Advanced Institute of Science and Technology.

Abstract

During the last decade most of coordinated security breaches are performed by the means of botnets, which is a large overlay network of compromised computers being controlled by remote botmaster. Due to high volumes of traffic to be analyzed, the challenge is posed by managing tradeoff between system scalability and accuracy. We propose a novel Hadoop-based P2P botnet detection method solving the problem of scalability and having high accuracy. Moreover, our approach is characterized not to require labeled data and applicable to encrypted traffic too.

I. Introduction

As an infrastructure for performing malicious activities, botnet can be used for distributed denial-of-service (DDoS) attacks, spamming, click fraud, identity theft, etc. Early botnets exhibited a centralized topology [2, 3], whereas more recent botnets [4, 8] started a topology shift into a peer-to-peer architecture. The main reason of this shift is a single point of failure in centralized C&C server architecture [5].

Recent advances in cloud computing and introduction of MapReduce [1] paradigm have been used in many data intensive computations. Certain advantage of Hadoop framework, an open-source version of MapReduce, is ability to execute tasks in distributed manner in a Hadoop Distributed File System (HDFS) [12] on commodity hardware. Moreover, it has its own recovery and fault-tolerance mechanisms. Our proposed method utilizes the advantages of Hadoop as well as behavioral flow analysis.

II. Related Work

Early botnet detection systems have been utilizing a numerous signature-based approaches [6]. A scalable signature-based approach was presented in Kargus [7] by accelerating signature matching in GPU.

Flow-based approach proposed in Gu et al. relies on the botnet lifecycle activities. Further, BotMiner [8] was proposed with the idea of correlating similar malicious activities with similar flows.

BotGraph [10] is one of the first applications having utilized the MapReduce

* {khalid.huseynov, kkj}@kaist.ac.kr

* This research was funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2014 [1391104001, Research on Communication Technology using Bio-inspired Algorithm]

*. This research was supported by the KUSTAR-KAIST Institute, KAIST, Korea.

paradigm in spamming botnet detection. BotCloud [11] is another detection method utilizing large graph processing capabilities of Hadoop. They have adapted the PageRank algorithm in the context of botnet detection and correlated the page rank of node with its probability of being bot.

III.Detection technique and methodology

3.1 Approach overview

The overview of the system architecture is shown in Fig.1. The Module 1 is used to parse *pcap* files in parallel in the HDFS. This library is adopted from the work of Lee et al. [13].

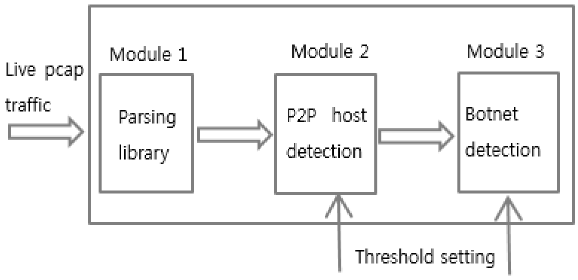

Fig. 1. Overview of the system architecture.

3.2 P2P host detection and implementation

The main purpose of Module 2 in Fig. 1 is to detect hosts with any kind of P2P activity. In order to differentiate P2P applications from normal user behavior (*e.g.* browsing, file downloads), we consider a number of features listed below.

Failed Connections. Normally, P2P applications expose higher number of failed connections due to the *peer churn* [14] phenomenon. We consider as failed any TCP or UDP flow with outgoing packet but no response packet, and a TCP flow with a reset packet.

Unresolved connections. DNS utilization behavior of P2P applications is different from the one of normal traffic [15]. Hosts running P2P applications resolve the IP list from the peers as opposed to DNS query. Thus we consider the number of DNS queries (answers) sent (received) as well as whether the flow have been previously resolved from DNS answer.

Destination subnet diversity. Another distinction of P2P traffic from normal Internet traffic is the diversity of destination hosts. Usually those hosts are scattered around numerous subnets separated geographically. Thus, we extracted the following two features: number of distinct IPs contacted by the host, and the number of different /16 prefix subnets connected by the host. Fig. 2 represents detailed design of Module 2 (P2P host detection) in Hadoop framework.

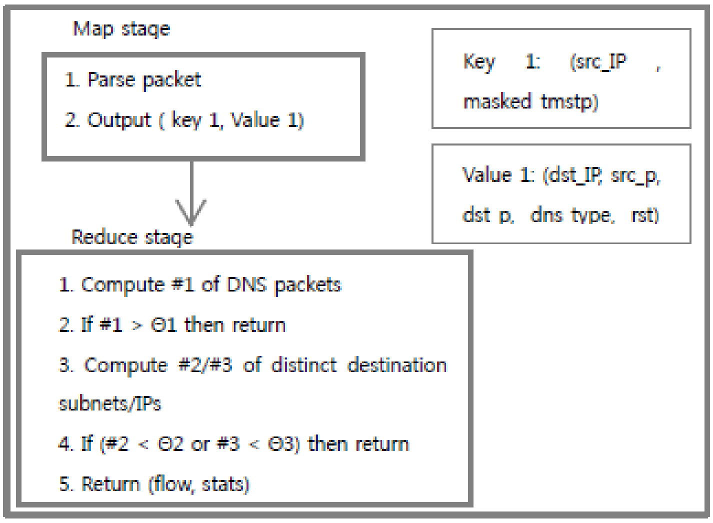

Fig.2 Design of Module 2 in Hadoop framework.

3.3 Fine-grained P2P botnet detection and implementation

Only traffic from the hosts running P2P applications is passed further to Module 3 in Fig. 1. The duty of Module 3 is to differentiate between hosts running legal P2P applications (*e.g.* Skype. eMule) and botnet infected ones. Here we utilize the following observations.

First of all, we use the observation that

P2P botnets have persistent flows compared to normal P2P applications [14].

Second observation is that the set of hosts connected by bots have more common hosts compared to legal P2P application.

ISOT dataset created by Information Security and Object Technology (ISOT) research lab at the University of Victoria was used for the benchmark [16]. Additionally we used a dataset consisting of legal P2P applications from the research group at Georgia Tech [9].

IV. Results and Discussion

Table 1 shows the detection results for Modules 2, and 3 of the detection system.

Table 1. Detection results

Type of P2P host	Module 2 (#detected/#total)	Module 3 (#botnet/ #P2P host)
Skype	6/7	1/6
eMule	2/2	0/2
Vuze	2/2	0/2
FrostWire	2/2	0/2
uTorrent	2/2	0/2
Storm botnet	13/13	12/13
Waledac botnet	3/3	3/3
Total	30/31	P2P bot: 15/16 Legal P2P: 1/14

In Module 2, we detect all kinds of P2P hosts. Detection includes legitimate as well as malicious P2P hosts. The results of this stage have only one host running Skype not detected as P2P (false negative). Other P2P hosts are detected with 100% accuracy. Thus, overall accuracy of this stage is 96.8% (30 out of 31 hosts).

In Module 3, using heuristics from Section 3.3, we differentiate between legal P2P hosts and P2P hosts running bot code. As you can see, almost all hosts running Storm or Waledac bot code have been identified correctly with accuracy of 93.7% (15/16). Furthermore, one legal P2P host running Skype application was misclassified as malicious host with false positive rate of 7% (1/14).

Table 2 shows our threshold settings for our implementation.

Table 2. Our threshold settings

Threshold	Corresponding feature	Threshold value
Θ1	DNS packets	<5
Θ2	Distinct destin. subnets	>=100
Θ3	Distinct destin. IPs	>=600
Θ4	$\frac{\text{runtime}}{\text{capturetime}}$	0.6
Θ5	$\frac{\text{matching IPs}}{\text{Total IPs}}$	0.7

Θ1 represents the number of any DNS packets exchanged during 10 minutes time interval. Θ2 represents the number of distinct destination subnets. Furthermore, Θ3 represents the number of distinct destination IP addresses. The last two features are introduced to differentiate between normal P2P and malicious P2P traffic. Θ4 set to 0.6 means that botnets exhibit communication more than 60% of the capture time. Moreover, Θ5 set to 0.7 means that 70% of destination IPs are same within the botnet.

Note that these thresholds are targeted to be set by network administrator.

V. Conclusion and Future work

Our contribution from this work can be described from multiple perspectives. First of all, we have developed unsupervised method for botnet detection, meaning we do not require any labeled data for training the system. Secondly, the accuracy of the system can be compared to the state-of-the-art detection methods. Furthermore, threshold setting makes it customizable for network administrators. Lastly, our system is inherently scalable due to development in Hadoop environment.

In the future work we plan to extend the system into application profiling framework. Also benchmark of the system on large volume dataset in a cluster environment is required.

References

[1] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.

[2] C. Miller, (2008). The Rustock Botnet Spams Again.

[3] Stone-Gross, B., Cova, M., Gilbert, B., Kemmerer, R., Kruegel, C., & Vigna, G. (2011). Analysis of a botnet takeover. Security & Privacy, IEEE, 9(1), 64-72.

[4] Stover, S., Dittrich, D., Hernandez, J., & Dietrich, S. (2007). Analysis of the Storm and Nugache Trojans: P2P is here. USENIX; login, 32(6), 18-27.

[5] Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., ... & Vigna, G. (2009, November). 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). ACM.

[6] Goebel, J., & Holz, T. (2007, April). Rishi: Identify bot contaminated hosts by irc nickname evaluation. In Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (pp. 8-8).

[7] Jamshed, M. A., Lee, J., Moon, S., Yun, I., Kim, D., Lee, S., ... & Park, K. (2012, October). Kargus: a highly-scalable software-based intrusion detection system. In Proceedings of the 2012 ACM conference on Computer and communications security (pp. 317-328). ACM.

[8] Gu, G., Perdisci, R., Zhang, J., & Lee, W. (2008, July). BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection. In USENIX Security Symposium (pp. 139-154).

[9] Rahbarinia, B., Perdisci, R., Lanzi, A., & Li, K. (2014). Peerrush: Mining for unwanted p2p traffic. Journal of Information Security and Applications.

[10] Zhao, Y., Xie, Y., Yu, F., Ke, Q., Yu, Y., Chen, Y., & Gillum, E. (2009, April). BotGraph: Large Scale Spamming Botnet Detection. In NSDI (Vol. 9, pp. 321-334).

[11] Francois, J., Wang, S., Bronzi, W., State, R., & Engel, T. (2011, November). BotCloud: detecting botnets using MapReduce. In Information Forensics and Security (WIFS), 2011 IEEE International Workshop on (pp. 1-6). IEEE.

[12] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on (pp. 1-10). IEEE.

[13] Lee, Y., & Lee, Y. (2013). Toward scalable internet traffic measurement and analysis with hadoop. ACM SIGCOMM Computer Communication Review, 43(1), 5-13.

[14] Stutzbach, D., & Rejaie, R. (2006, October). Understanding churn in peer-to-peer networks. In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (pp. 189-202). ACM.

[15] Wu, H. S., Huang, N. F., & Lin, G. H. (2009, June). Identifying the use of data/voice/video-based P2P traffic by DNS-query behavior. In Communications, 2009. ICC'09. IEEE International Conference on (pp. 1-5). IEEE.

[16] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., & Garant, D. (2013). Botnet detection based on traffic behavior analysis and flow intervals. Computers & Security, 39, 2-16.