Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.4, Aug. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.4.807

하둡 프레임워크에서 한계점 가변으로 확장성이 가능한 P2P 봇넷 탐지 기법*

Khalid Huseynov,[1†] Paul D. Yoo,[2] 김 광 조[1‡]
[1]한국과학기술원, [2]Bournemouth University (UK)

Scalable P2P Botnet Detection with Threshold Setting in Hadoop Framework*

Khalid Huseynov,[1†] Paul D. Yoo,[2] Kwangjo Kim[1‡]
[1]Korea Advanced Institute of Science and Technology (KAIST),
[2]Bournemouth University

요 약

최근 10년 전부터 대부분의 조직화된 사이버 공격은 원격의 Botmaster에 의해 통제되는 감염된 컴퓨터들의 거대한 네트워크인 Botnet을 통해 이루어졌다. Botnet 탐지는 상당한 양의 네트워크 트래픽 분석이 필요하기 때문에 탐지 정확도와 시스템 확장성간의 적절한 타협이 요구된다. 본 연구에서는 높은 탐지율을 제공하면서 시스템 확장성이 가능한 하둡 기반의 새로운 P2P Botnet 탐지 기법을 제안한다. 또한, 본 제안 기법은 레이블이 되지 아니한 공격 데이터 뿐만 아니라 암호화된 공격 트래픽에도 적용이 가능한 특징을 가지고 있다.

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 as well.

Keywords: botnets, scalability, Hadoop, unsupervised detection.

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. Infected hosts are controlled by C&C server in either direct (centralized) or indirect (P2P) way, depending on overlay network topology [1,2]. Early botnets exhibited a centralized topology [3-5], whereas more recent botnets [6,7] started a topology shift into peer-to-peer

접수일(2015년 3월 25일), 수정일(2015년 7월 23일), 게재확정일(2015년 7월 28일)
* This work was partly supported by the ICT R&D program of MSIP/IITP, Republic of Korea. [1391104001, Research on Communication Technology using Bio-inspired Algorithm] and KUSTAR-KAIST Institute, KAIST, Korea.
† 주저자, khalidhnv@gmail.com
‡ 교신저자, kkj@kaist.ac.kr(Corresponding author)

(P2P) architecture. The main reason of this shift is a single point of failure in centralized C&C server architecture [8]. Meanwhile, P2P botnets utilize distributed communication protocols [1,2] making it resilient against takeover [9].

Early botnet detection methods were based on precomputed signatures that could be embedded into Intrusion Detection Systems (IDSs) [10,11]. However, previous method does not scale well, and signatures should be devised from scratch for any new type of attack/botnet[12]. Moreover, the attackers can easily evade some of detection signatures by encrypting the payload [13]. Further methods have been focused more on a flow level behavior of botnets [14,15,16]. Moreover, flow-based detection techniques are applicable to encrypted traffic as well due to unnecessity in packet load inspection. Also, application of statistical and machine learning techniques are useful in this flow-wise behavior analysis[17,18].

Recent advances in the distributed cluster computing and the introduction of MapReduce [19] paradigm have been useful in a number of data intensive tasks. An open source version of MapReduce, namely Hadoop framework, has led to even higher acceptance by research community. One of the advantages of Hadoop framework includes its ability to distribute and execute tasks in a distributed manner in a Hadoop Distributed File System (HDFS) [27] on commodity hardware. Moreover, Hadoop has its own recovery and fault-tolerance mechanisms.

Our proposed method utilizes the advantages of Hadoop as well as behavioral flow analysis. This framework is particularly useful in the case of P2P traffic analysis due to inherent flow characteristics of this type of applications. The proposed contributions are as follows.

- ◉ Development of scalable detection method able to handle large volumes of data with high accuracy.
- ◉ No prior need for labeled data set.
- ◉ Possibility of detection for novel P2P botnets.

II. Related work

Early botnet detection systems have been utilizing numerous types of signature-based approaches. Rishi [10] is an example of signature-based detection scheme for botnets manipulated via IRC channels. A part of bot nickname has been used as a signature for identifying malicious packets. Further, the signature can be converted into IDS specific rule. Snort [11] and Bro [20] are examples of the state-of-the-art IDSs. A scalable signature-based approach was presented in Kargus [21] by accelerating signature matching in GPU.

On the other hand, flow-based anomaly detection methods are more flexible in terms of novelty detection. Gu et al. proposed BotHunter [14] that relies on botnet lifecycle activities. Those lifecycle activities include port scanning, infection, binary download, and C&C scanning. BotHunter considers this sequence of events as anomaly. Further, BotMiner [15] was proposed with the idea of correlating similar malicious activities with similar flows. Many of the state-of-the-art approaches utilize Machine Learning (ML) algorithms in tandem with the novel feature engineering techniques [16,22,23].

With the advent of the open source distributed computing frameworks such as Hadoop, the scalability issues of the

detection systems could be delegated to the underlying framework. BotGraph [24] is one of the first detection frameworks utilizing the MapReduce paradigm in spamming botnet detection. The main idea of the study is to find tightly connected sub-graph components in the constructed large user-user communication graph, which is a large graph with nodes representing users (hosts) and directed edges representing communication channels between them.

BotCloud [25] is another detection framework utilizing large graph processing abilities 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 infected. Furthermore, Singh et al. [27] proposed Hadoop-based analytics framework for botnet detection using supervised approach with random forests, which is a type of classification algorithms.

III. Detection technique and methodology

3.1 Approach overview

The overview of the system architecture is shown in Fig.1. Module 1 is used to parse pcap files, in parallel, directly from the HDFS. This library is adopted from the work of Lee et al. [28] and applies the heuristic based on similarity of timestamps of the consecutive packets.

Module 2, namely P2P host detection, is developed with the purpose of extracting traffic of any P2P applications, be it botnet or clean P2P traffic. Note that applications such as eMule, Bittorrent, or Skype utilize P2P protocols [1, 2] as well. Although protocols in [1, 2] are used less often recently, the new protocols behave similar to them from flow-wise point of view. Further, after detecting hosts running P2P applications of any kind, we can differentiate between the malicious and non-malicious P2P behavior in Module 3.

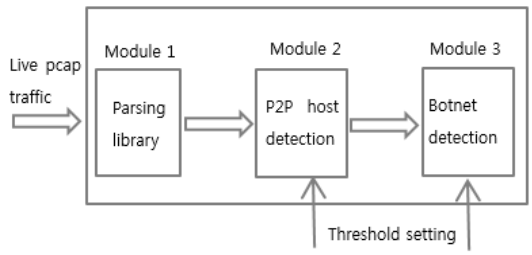

Fig. 1. Overview of the system architecture.

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 the normal user behavior (*e.g.*, browsing, file downloads, etc.), we consider a number of features listed in Table 1.

Failed Connections (Θ1). Normally, P2P applications expose higher number of failed connections due to the peer churn [29] phenomenon. We consider as failed any TCP or UDP flow with outgoing packet but no response packet, and a TCP flow with a reset bit set in the packet header.

Unresolved connections (Θ2). DNS utilization behavior of P2P applications is different from one of normal traffic [30]. 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 diversity (Θ3,Θ4). Another

Table 1. Feature descriptions.

Feature	Meaning	Granularity
Θ1	Failed connections	Host
Θ2	DNS packets	Host
Θ3	Distinct destin. subnets	Host
Θ4	Distinct destin. IPs	Host
Θ5	active time capture time	Flow
Θ6	overlapping IPs Total IPs	Host

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: the number of distinct IPs contacted by the host, and the number of different CIDR (Classless Inter-Domain Routing) prefix (/16) subnets connected by the host.

Fig. 2 represents detailed design of Module 2 (P2P host detection) in Hadoop framework. As you can see, Module 2 is implemented using one map and one reduce functions. The key of Map stage maps all the packets in *tmstp* period from *src_IP* into same Reduce stage function.

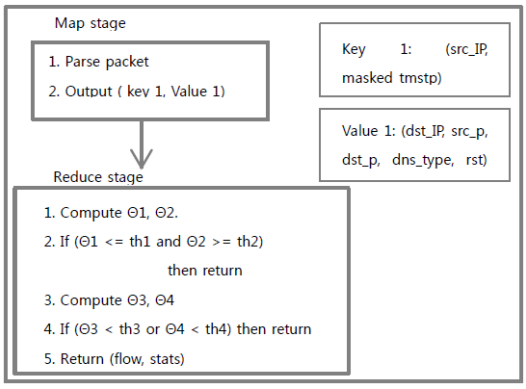

Fig. 2. Design of Module 2 in Hadoop framework

This way we can extract time-interval flows in Reduce stage.

Four main features (denoted as Θ1, Θ2, Θ3 and Θ4) are computed on reduce stage of this module. Furthermore, they are compared to the corresponding thresholds (denoted as "th"), and the hosts satisfying given conditions proceed to next step and are selected as P2P hosts.

The conditional statements in Reduce stage of Fig. 2 represent already mentioned heuristics. As an example, first clause (Θ1 〈= th1) of line 2 represents the case when number of failed connections is less than given threshold. In this case, flow is more likely to be a normal traffic rather than P2P traffic, leading to discarding of current flow interval from further evaluation. Similarly, second clause (Θ2 〉= th2) represents the case of large number of DNS resolutions that may not happen in case of P2P traffic.

Further, line 4 in Reduce stage of Fig. 2 represents the heuristic referring to destination diversity of hosts connected to botnet. Thus this conditional statement discards all the flows with the destination diversity less than mentioned threshold.

3.3 Fine-grained P2P botnet detection and implementation

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

First of all, we make use of the observation that P2P botnets have more persistent flows compared to normal P2P applications [29]. This means that the average lifespan of a flow created by

botnet is noticeably longer that the one created by legal P2P application. We can use our fifth feature (Θ5) to denote the lifespan of a flow as the ratio of the time when a flow is being active to the duration of the capture.

The second observation is that the set of hosts connected by bots have more overlap than the ones connected by legal P2P applications [17]. Thus we can set up the sixth feature (Θ6) as the ratio of overlap in destination IPs to the total number of destination IPs.

Module 3 has been implemented using Python script without involvement of Hadoop framework. The reason is that most of traffic intensive computations have been executed in Module 2, resulting in few orders smaller output from it. Thus Hadoop implementation of Module 3 would incur more overhead compared to Python script running on master node of a Hadoop cluster.

3.4 Benchmark dataset and cluster setup

One of our evaluation benchmarks is ISOT dataset that was created by Information Security and Object Technology (ISOT) research lab at the University of Victoria [31]. 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 [32] and includes Storm and Waledac botnets.

Non-malicious traffic was collected from two sources. One source is the Traffic Lab at Ericsson Research in Hungary [33]. This traffic was integrated with second dataset built by Lawrence Berkeley National Lab (LBNL) [34]. 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 bittorrent 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 traces.

Additionally we used a dataset consisting of legal P2P applications from the research group at Georgia Tech [16]. This dataset includes five P2P application traces, namely Skype, eMule, Vuze, FrostWire, and uTorrent. Two hosts are dedicated for each of those applications, except Skype. There are seven hosts running Skype with two of them selected as Skype supernodes.

For the sake of realistic evaluation, we have set up the five-node Hadoop cluster in our laboratory environment. The cluster includes one master node and four slave nodes. System configurations on all the nodes are set as follows:

◉ OS: Ubuntu 14.04 LTS 64-bit
◉ Kernel version: Linux 3.11.0-23-generic
◉ Hadoop version: Apache Hadoop 1.2.1

Another important setting for Hadoop cluster performance is the number of map and reduce tasks. By following the best practices, we set the number of map tasks approximately to the number of Hadoop file blocks, and the number of reduce tasks is set to ⌈ # map tasks / 10 ⌉.

IV. Results and discussion

Table 2 represents detection runtime results on our five-node Hadoop cluster. As you can see, overall of 67.3 GB of malicious as well as non-malicious P2P traffic and 11 GB of normal user traffic have been processed with the average

Table 2. Detection runtime results

Type of traffic	Size of traffic (GB)	Processing time (s)
Skype	6.5	140.5
eMule	19.9	391.6
Vuze	9.7	272.7
FrostWire	5.9	150
uTorrent	19.8	385.9
Waledac botnet	0.26	23.4
Zeus botnet	0.1	18.7
Storm botnet	5.1	162.4
Normal traffic	11	228.3
Total	78.3	1773.5
Average	1	22.6

speed of 22.6 second per 1 GB of pcap raw traffic. This time represents the most data-intensive component of the system, which was implemented in Module 2. Once the raw traffic is passed through Module 2, the traffic volume is dramatically decreased (<10 MB text per 1 GB of raw traffic). Further, additional overhead of at most 3 seconds is incurred by Module 3.

Thus, it takes at most 25.6 seconds for detecting any botnet infected hosts from the 1 GB of raw network traffic. Table 3 shows comparison with another botnet detection framework that utilizes supervised approach [26]. As you can see, our unsupervised detection system twice outperforms it due to lower number of features and mostly focus on host-wise detection as opposed to flow-wise. Moreover, model training and classification itself, in case of supervised approach, add additional overhead.

In addition to providing scalability, our

Table 3. Runtime performance comparison

Method	Total processing time (s)
Our approach	25.6
K. Singh et al. [26]	63

Table 4. Average feature values and our threshold settings

Feature	P2P traffic		Normaltraffic	Threshold value
	Malicious	Non-malicious		
Θ1	30.5	11.6	1.9	>=5
Θ2	1.2	1.6	5.3	<=3
Θ3	439.6	788.3	2.9	>=45
Θ4	521.6	929.7	3.2	>=60
Θ5	0.941	0.716	-	>0.82
Θ6	0.91	0.55	-	>0.73

second goal is to provide easy threshold setting with high accuracy of detection. Table 4 presents the average feature values that were obtained after the analysis of traffic from malicious and non-malicious P2P applications, as well as normal user traffic.

As previously discussed, Θ1 denotes the number of failed connections during the chosen time interval, which is 10 minute in our case. Thus, we can observe that P2P applications have an order of magnitude more disconnections as opposed to normal user behaviour. Θ2 represents the number of any DNS packets exchanged during the same time interval. For example, P2P bots send/receive on average 1.2 DNS requests/replies every 10 minutes. This implies that new bot may join every 17 minute with the DNS lookup for the well-known "introducer" server. Furthermore, Θ3 and Θ4 represent the destination address diversity. This is yet another highly distinguishing feature with two orders of magnitude higher values compared to normal user traffic. Thus, the mentioned four features have been used to differentiate normal traffic from P2P traffic, and the threshold values are mentioned in the last column.

The last two features are introduced to

differentiate between non-malicious and malicious P2P traffics. Θ5 set to 0.82 means that the persistent P2P flows exhibiting communication more than 82% of the capture time are considered to be malicious. Moreover, Θ6 set to 0.73 means that hosts with an overlap of more than 73% in destination IPs are considered to belong to the same botnet.

Finally, while utilizing all the threshold values from the last column of Table 4, we obtained the detection results as shown in Table 5. The detection results for Modules 2 and 3 are shown in second and third columns of Table 5, respectively.

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 96.9% (31 out of 32 hosts).

In Module 3, using heuristics from Section 3.3, we differentiate between legal P2P hosts and the infected ones. From the total standings of Table 5, all infected hosts running bot code have been identified correctly with true positive (TP) of 16 out of 17 and the corresponding accuracy of 94.1%. Furthermore, one legal P2P host running Skype application was misclassified as malicious host with false alarm (FA) rate of 3.1% (1/32).

Table 5. 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
Zeus botnet	1/1	1/1
Waledac botnet	3/3	3/3
Storm botnet	13/13	12/13
Total	31/32	TP: 16/17 FA: 1/32

Table 6. Comparison of detection results

Approach	True positive rate	False alarm rate
Ours	94.1%	3.1%
K. Singh et al. [26]	99.8%	0.3%
BotGraph [24]	>95%	0.44%
Bot Cloud [25]	99%	3%

Comparison with other methods shows little lack in accuracy. This is due to naive threshold setting approach. Currently our threshold is set as the average of P2P and normal traffic parameters. More advanced threshold setting can give improvement in accuracy as well as adoption to different botnet models.

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.

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

References

[1] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," In Peer-to-Peer Systems, pp. 53-65, Springer Berlin Heidelberg, Jan. 2002.

[2] M. Ripeanu, "Peer-to-peer architectu- re case study: Gnutella network," In Peer-to-Peer Computing Proceedings. First International Conference on, pp. 99-100, IEEE, Aug. 2001.

[3] G. Keizer, "Top botnets control 1 M hijacked computers," Apr. 2008. www.computerworld.com/article/2536378/security0/top-botnets-control-1m-hijacked-computers.html

[4] C. Miller, "The Rustock Botnet Spams Again," 2008.

[5] B. Stone-Gross, M. Cova, B. Gilbert, R. Kemmerer, C. Kruegel, and G. Vigna, "Analysis of a botnet takeover," Security and Privacy, IEEE, vol. 9, no. 1, pp. 64-72, 2011.

[6] D.I. Jang, M. Kim, H.C. Jung, and B.N. Noh, "Analysis of HTTP2P botnet: case study waledac," In Communications (MICC), IEEE 9th Malaysia International Conference on, pp. 409-412, IEEE, Dec. 2009.

[7] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich, "Analysis of the Storm and Nugache Trojans: P2P is here," USENIX: login, vol. 32, no. 6, pp. 18-27, 2007.

[8] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, ... 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, ACM, Nov. 2009.

[9] C. Rossow, D. Andriesse, T. Werner, B. Stone-Gross, D. Plohmann, C.J. Dietrich, and H. Bos, "Sok: P2pwned-modeling and evaluating the resilience of peer-to-peer botnets," In Security and Privacy (SP), IEEE Symposium on, pp. 97-111, IEEE, May 2013 .

[10] J. Goebel and T. Holz, "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, April 2007.

[11] M. Roesch, "Snort: Lightweight Intrusion Detection for Networks," In LISA, vol. 99, no. 1, pp. 229-238, Nov. 1999.

[12] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming botnets: signatures and characteristics." In ACM SIGCOMM Computer Communication Review, vol. 38, no. 4, pp. 171-182, ACM, Aug. 2008.

[13] E. Stinson and J.C. Mitchell, "Towards Systematic Evaluation of the Evadability of Bot/Botnet Detection Methods," USENIX Workshop on Offensive Technologies (WOOT), vol. 8, pp. 1-9, 2008.

[14] G. Gu, P.A. Porras, V. Yegneswaran, M.W. Fong, and W. Lee, "BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation," In Usenix Security, vol. 7, pp. 1-16, Aug. 2007.

[15] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection." In USENIX Security

Symposium, vol. 5, no. 2, pp. 139-154, July 2008.

[16] B. Rahbarinia, R. Perdisci, A. Lanzi, and K. Li. "Peerrush: Mining for unwanted p2p traffic," Journal of Information Security and Applications, vol. 19 no. 3, pp. 194-208, 2014.

[17] J. Zhang, R. Perdisci, W. Lee, X. Luo, and U. Sarfraz, "Building a scalable system for stealthy p2p-botnet detection." Information Forensics and Security, IEEE Transactions on, vol. 9, no. 1, pp. 27-38, 2014.

[18] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, ... and P. Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," In Privacy, Security and Trust (PST), Ninth Annual International Conference on, pp. 174-180, IEEE, July 2011.

[19] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107-113, 2008.

[20] V. Paxson, "Bro: a system for detecting network intruders in real-time," Computer networks, vol. 31, no. 23, pp. 2435-2463, 1999.

[21] M.A. Jamshed, J. Lee, S. Moon, I. Yun, D. Kim, S. Lee, ... and K. Park, "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, Oct. 2012.

[22] B. Wang, Z. Li, H. Tu, & J. Ma, "Measuring peer-to-peer botnets using control flow stability," In Availability, Reliability and Security, ARES'09, International Conference on, pp. 663-669, IEEE, March 2009.

[23] D. Zhao, I. Traore, A. Ghorbani, B. Sayed, S. Saad, and W. Lu, "Peer to peer botnet detection based on flow intervals," In Information Security and Privacy Research, pp. 87-102, Springer Berlin Heidelberg, 2012.

[24] Y. Zhao, Y. Xie, F. Yu, Q. Ke, Y. Yu, Y. Chen, and E. Gillum, "BotGraph: Large Scale Spamming Botnet Detection." In NSDI, vol. 9, pp. 321-334, April 2009.

[25] J. Francois, S. Wang, W. Bronzi, R. State, and T. Engel, "Botcloud: Detecting botnets using mapreduce." In Information Forensics and Security (WIFS), IEEE International Workshop on, pp. 1-6, IEEE, Nov. 2011.

[26] K. Singh, S.C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forests," Information Sciences, vol. 278, pp. 488-497, 2014.

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

[28] Y. Lee and Y. Lee. "Toward scalable internet traffic measurement and analysis with hadoop," ACM SIGCOMM Computer Communication Review, vol. 43, no. 1, pp. 5-13, 2013.

[29] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks." In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, pp. 189-202, ACM, Oct. 2006.

[30] H.S. Wu, N.F. Huang, and G.H. Lin, "Identifying the use of data/voice/video-based p2p traffic by dns-query behavior," In Communications, ICC'09. IEEE International Conference on. pp. 1-5, IEEE, June 2009.

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

2013.
〔32〕 French Chapter of Honeynet, November, Nov. 2014.www.honeynet.org/chapters/france
〔33〕 G. Szabó, D. Orincsay, S. Malomsoky, and I. Szabó, "On the validation of traffic classification algorithms," In Passive and Active Network Measurement, pp. 72-81. Springer Berlin Heidelberg, 2008.
〔34〕 LBNL Enterprise Trace Repository, November 2014. www.icir.org/enterprise-tracing

〈저자소개〉

Khalid Huseynov 학생회원
2013년 2월: 한국과학기술원 전산학부 졸업
2015년 2월: 한국과학기술원 전산학부 석사
2015년 2월~2015년 4월: 한국과학기술원 전산학부 암호와 정보보안 연구실에서 연구원
2015년 4월~현재: R&D Software Engineer at NFLabs
〈관심분야〉 네트워크보안, 봇넷, 트래픽 분석, 빅 데이터

Paul D. Yoo 정회원
2008년 2월: Received PhD in Engineering and IT from the University of Sydney, Australia
2008년 12월~2009년 8월: Research Fellow, Distributed and High Performance Computing, USYD, Australia
2009년 8월~2015년 1월: Assistant Professor, ECE Dept. and ATIC-Khalifa Semidonductor Research Centre, KUSTAR, UAE
2015년 1월~현재: Lecturer (equiv. Assistant Professor), Data Science Institute, Bournemouth University, United Kingdom
〈관심분야〉 data science, informatics, cyber security

김 광 조 (Kwangjo Kim) 종신회원
1980년 2월: 연세대학교 공과대학 전자공학과 졸업
1983년 8월: 연세대학교 대학원 전자공학과 석사 (M/W 전공)
1991년 3월: 일본 요코하마 국립대 대학원 전자정보공학 박사 (암호학 및 정보보호 전공)
1979년 12월~1997년 12월: 국가보안기술연구소 부호1실장/책임연구원
1999년 1월~2004년 12월: 세계암호학회(IACR) 이사
1998년 1월~2009년 2월: 한국정보통신대학교 정보통신대학원장 및 공학부장
2003년 1월~2005년 1월: IT 영재교육원 원장
2005년 1월~2008년 12월: Asiacrypt 조정위원회 의장
2005년 2월~2005년 5월: MIT 방문학자
2005년 6월~2005년 11월: UCSD 방문교수
2009년 3월~2009년 12월: 한국정보보호학회 회장
2012년 1월~2012년 8월: KUSTAR(UAE) 방문교수
2013년 1(7)월~2013년 2(8)월: ITB(인도네시아) 방문교수
2009년 3월~현재: 한국과학기술원 전산학부 교수
2010년 1월~현재: 한국정보보호학회 명예회장
2014년 4월~현재: IFIP TC -11 한국대표
〈관심분야〉 암호와 정보보호 이론 및 응용