Information Sciences 511 (2020) 284–296

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors

Wei Wang[a,b], Yaoyao Shang[a,b], Yongzhong He[a,b], Yidong Li[a,b,*], Jiqiang Liu[a,b]

[a] Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, 3 Shangyuancun, Beijing 100044, China

[b] School of Computer and Information Technology, Beijing Jiaotong University, 3 Shangyuancun, Beijing 100044, China

ARTICLE INFO

Article history:
Received 15 July 2018
Revised 10 August 2019
Accepted 13 September 2019
Available online 20 September 2019

Keywords:
Botnet detection
Network security
Intrusion detection
Network monitoring
Machine learning

ABSTRACT

The Botnets have become one of the most serious threats to cyber infrastructure. Most existing work on detecting botnets is based on flow-based traffic analysis by mining their communication patterns. There also exists related work based on anomaly detection in communication graphs. As bots have continuously evolved and become increasingly sophisticated, only using flow-based traffic analysis or graph-based analysis for the detection would result in false negatives or false positives, or can even be evaded. In this work, we propose BotMark, an automated model that detects botnets with hybrid analysis of flow-based and graph-based network traffic behaviors. We extract 15 statistical flow-based traffic features as well as 3 graph-based features in building the detection model. For flow-based detection, we consider the similarity and stability of C-flow as measurements in the detection. In particular, we employ *k*-means to measure the similarity of C-flows and assign similarity scores, and calculate stability score of C-flows through the distribution of packet length within a C-flow. The graph-based detection is based on the observation that the neighborhoods of anomalous nodes significantly differ from those of normal nodes in communication graphs. In particular, we use least-square technique and Local Outlier Factor (LOF) to calculate anomaly scores that measure the differences of their neighborhoods. Our models use the scores to mark bots. BotMark performs automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors by ensemble of the detection results based on similarity scores, stability scores and anomaly scores. We collect a very large size of network traffic by simulating 5 newly propagated botnets, including Mirai, Black energy, Zeus, Athena and Ares in a real computing environment. Extensive experimental results demonstrate the effectiveness of BotMark. It achieves 99.94% in terms of detection accuracy, outperforming any individual detector with flow-based detection or graph-based detection.

1. Introduction

A botnet is formed by a large number of hosts that are infected with zombie programs. Bots can be remotely controlled by attackers for Distributed Denial of Service (DDoS) attacks, for spreading spams, for conducting click fraud or for stealing

* Corresponding author at: School of Computer and Information Technology, Beijing Jiaotong University, 3 Shangyuancun, Beijing 100044, China.
E-mail addresses: wangwei1@bjtu.edu.cn (W. Wang), 16120335@bjtu.edu.cn (Y. Shang), yzhhe@bjtu.edu.cn (Y. He), ydli@bjtu.edu.cn (Y. Li), jqliu@bjtu.edu.cn (J. Liu).

https://doi.org/10.1016/j.ins.2019.09.024
0020-0255/

personal information. When a botmaster commands a botnet, it requires a Command and Control (C&C) channel to accomplish such as scanning, binary download or other suspicious activities. The Internet Relay Chat (IRC) protocol was considered as one of the most popular botnet communication protocols employing the centralized topology. Botmasters usually build private channels and hide themselves by encryption. Later, a HTTP-based botnet emerged. HTTP-based botnets are more difficult to be detected since the http packets generated by the bots can be flood in large amounts of web traffic records. Obviously, there exist vulnerabilities in IRC and HTTP-based botnets. Once C&C server is identified and then closed, the entire botnet will be destroyed. In order to evade the identification, botmasters began to develop P2P (Peer to Peer) based botnets with strong concealment and robustness.

Botnet detection has been a widely studied topic and many approaches [5,8,11,15,18–21,24,26,41] have been proposed. In general, the signature-based approaches [26] are unable to detect unknown botnets and their variants, or even fail in case of encryption. The anomaly-based detection approaches [8,15,21,41] are based on the assumption that the communication pattern of botnets is different from those of benign hosts in the networks. The limitations of anomaly-based detection approaches are that bots may mimic the communication patterns of normal hosts to evade the detection. The detection approaches based on honeypot technology can only detect existing bots, and has poor real-time performance. The detection approaches based on specific protocols and structures [5,15,19,20,26] are unable to detect botnets with different protocols or structures. The community-based anomaly detection algorithms [11,18,24] cannot accurately identify botnets when full communication graphs are unavailable.

Bots have been evolving quickly and become increasingly sophisticated. The existing botnet detection approaches may become ineffective against the novel botnets or their variants. In order to effectively characterize the behaviors of bots and thus more accurately detect novel botnets, in this paper, we extend our previous work [22], and propose an automated detection model called BotMark for the detection of botnets with hybrid analysis of flow-based and graph-based traffic behaviors. In flow-based traffic analysis, we extract 15 statistical features including duration, the number of packets within an flow and total length of packets, etc. We consider the similarity and stability of C-flow as measurements in the detection. In particular, we employ k-means to measure the similarity of C-flows and assign similarity scores, and calculate stability score of C-flows through the distribution of packet length within a C-flow. In graph-based traffic analysis, we extract 3 types of graph-based features such as the number of nodes, the number of edges and weights in egonet. The graph-based detection is based on the observation that the neighborhoods of anomalous nodes significantly differ from those of normal nodes in communication graphs. In particular, we use least-square technique and Local Outlier Factor (LOF) to calculate an anomaly score that measures the differences of their neighborhoods. Our models use the scores to mark bots. BotMark performs automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors by ensemble of the detection results based on similarity scores, stability scores and anomaly scores.

We collect a very large size of network traffic by simulating 5 newly propagated botnets, including Mirai, Black energy, Zeus, Athena and Ares in a real computing environment. Extensive experimental results demonstrate the effectiveness of BotMark. It achieves the detection accuracy as 99.49% with flow-based detector and 91.66% with graph-based detector. BotMark reaches the detection accuracy of 99.94% with hybrid of both detectors, outperforming any individual detector. The experimental results also show that the flow-based detector are more effective and efficient for botnet detection than the graph-based detector. However, the graph-based method is able to detect part of Zeus bots that cannot be identified by C-flow based detector, and they well make up for the deficiency from patterns of node neighborhoods aspect. It is clear that the hybrid analysis of flow-based and graph-based traffic behaviors is meaningful and necessary to detect the botnets.

We make the following contributions:

(1) We propose BotMark that automatically detects bots with hybrid analysis of flow-based and graph-based traffic behaviors. BotMark can characterize the botnets' behaviors more comprehensively than any individual analysis. Moreover, Botmark is independent of botnet C&C protocol and structure, requires no *a priori* knowledge of botnets, and thus can be adopted in complex networking environments.
(2) We extract as many as 15 statistical flow-based features and 3 types of graph-based features from network traffic to comprehensively characterize the behaviors of botnets. We mine the similarity and stability of botnet communication patterns in the flow-based detector. We further set up patterns of normal node neighborhoods in the graph-based detector. We compare the detection performance with both types of detectors and make final hybrid analysis.
(3) We collect a very large size of network traffic by simulating 5 newly propagated botnets, including Mirai, Black energy, Zeus, Athena and Ares in a real computing environment. We share our data in the research community. Extensive experimental results demonstrate the effectiveness of BotMark. It reaches the detection accuracy of 99.94% with hybrid detectors, outperforming any individual detector.

The remainder of this paper is organized as follows. We review related work in Section 2. Section 3 introduces BotMark, and explains both detectors in details. Section 4 describes the data sets and experiments. We discuss the limitations of the method in Section 5. Section 6 concludes this paper.

2. Related work

Many botnet detection approaches have been proposed in recent years. Livadas et al. [17] proposed supervised machine learning based classification techniques on statistical flow characteristics to identify bots. They distinguished malicious traffic

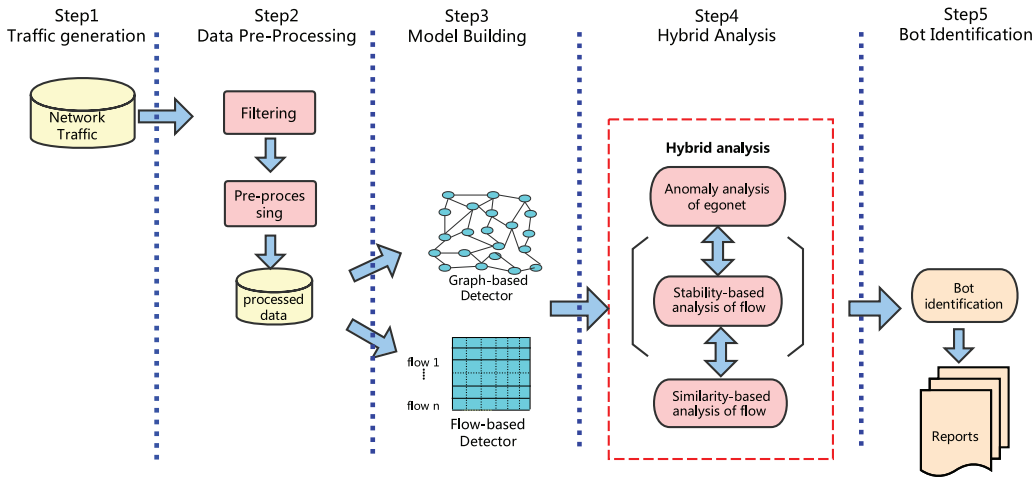

Fig. 1. Overview of BotMark.

generated by IRC bots from the traffic generated by normal hosts in the first stage, and then extracted flow characteristics and classified flows into malicious or non-malicious in the second stage. Beigi et al. [21] proposed group exclusion and feature inclusion to select effective features. In particular, they employed C4.5 (decision tree) with reduced error pruning algorithm (REP) to classify the traffic flows. Gu et al. [8] designed a system called BotMiner that exploits spatial and temporal characteristics of botnet traffic to identify bots. In our previous work, we also use machine learning methods for the detection of malicious Android Applications [31,33,36–38] or intrusions in computer networks [27,29,30,32] or in computer systems [28,35]. We also try to mitigate privacy risk in Android [14] or in Blockchain [16].

Apart from specific botnet detection algorithms, there also exist related work that employs graphs to identify the presence of botnets [1,2,4,6,7,9,10,13,25,39]. Sudipta et al. [6] proposed a method based on topological features of nodes within a graph. They enhanced the efficiency by removing inactive nodes. Sofiane et al. [13] developed a graph based models of NetFlow records which focused on modeling dependencies among flows, namely BotGM. There also exists general methods for identifying communities in networks, e.g., modularity optimization [4] or graph clustering [39]. Wang et al. [25] analyzed the social relationship between nodes by constructing SIG and SCG to distinguish botnet communities. The authors used large deviation on the degree distribution in social interaction graphs to find pivotal nodes. They then detected nodes that interact highly with pivotal nodes. Iliofotou et al. [10] identified P2P flows by calculating the hosts ratio of in degree to out degree in protocol traffic graphs. Francois et al. [7] presented BotTrack. They employed PageRank algorithm with an additional clustering process to identify groups of hosts sharing similar communication patterns. In addition, some researchers have incorporated these graph metrics in curve fitting approaches to detect anomalies in large scale graphs. Akoglu et al. [1] proposed an Oddball algorithm that utilized a number of power-laws to detect anomalies. Raza et al. [9] proposed lots of metrics based on undigraph to detect anomalies, i.e., average betweenness centrality and community cohesiveness.

The existing work used either flow-based method or graph based method to detect botnets. However, bots has been quickly evolving and botnets have become increasingly sophisticated. Moreover, botmasters may utilize techniques such as flow pertubance for avoiding detection. The individual detection method may thus be unable to detect botnets.

In our previous work [22], we detected botnets with hybrid analysis of flow-based and graph-based features of network traffic. The method requires to label the training data. However, there are not enough well-labeled data sets to train the models in the real world. Moreover, due to the quick evolution of bots, training models based on known botnets may not be suitable for the detection of novel bots or their variants. In this work, an important advantage of BotMark is that it does not need labeled data and thus can work well for tasks that do not have enough labeled data. We also [34] proposed a botnet detection method based on two-layered analysis with graph anomaly detection and network traffic clustering, namely BotCapturer in computing environments with no enough labeled data. However, the graph-based method is only used to filter flows in order to reduce the traffic workload in the first stage. In this work, we propose BotMark that performs automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors by ensemble of the detection results based on similarity scores, stability scores and anomaly scores.

3. Method

We illustrate BotMark in Fig. 1. BotMark works in five steps: traffic generation, data preprocessing, modeling building, hybrid analysis and bot identification. In the first step, we collect a very large size of network traffic by simulating five newly propagated botnets and mix background traffic generated from a telecom company. In the second step, we filter out irrelevant traffic flows in order to reduce the network traffic workload. In the third step, we extract 15 statistical flow-based features as well as 3 graph based features from network traffic to characterize the behavior of botnets. We perform hybrid

Table 1
Descriptions of traffic flows features.

Feature	Description	Appeared in Reference
SrcIp	Source IP address	
SrcPort	Source port address	
DstIp	Destination IP address	
DstPort	Destination port address	
Duration	Flow duration	[3]
PX	Total number of transmitted packets	[3,20,21,23,41]
NSP	Number of small packets (length of 63–400 bytes)	[3,15]
AIT	Average arrival time of packets	[20]
TBT	Total number of transmitted bytes	[3,15,20,23]
APL	Average payload packet length for time interval	[3,8,21,23,41]
PV	Standard deviation of payload packet length	[3,23,41]
FPS	The size of the first packet in the flow	[3,12,21,41]
DPL	The total of number of different packet size over the total number of packets	[21]
MPL	The maximum of packet length in the flow	[23]
MP	The number of maximum packets	[23]
MB	The total number of bytes transmitted by the largest packet	[23]
BPS	The average bits-per-second	[8,20,40]
PPS	The number of packets per second	[3,8,20]
FPH	The number of C-flows per hour	[3,8,20]

analysis of flow based and graph based features of traffic behaviors in the fourth step. In the final step, we validate the effectiveness with extensive experiments of BotMark, including: (1) compare the performance of similarity-based detector, stability-based detector and their hybrid analysis; (2) analyze the detection performance of graph detector; (3) performance hybrid analysis by ensemble of the detection results obtained with the flow-based detector and graph-based detector.

3.1. Traffic pre-processing

In order to reduce traffic workload and improve efficiency, we filter out irrelevant traffic flows. We firstly filter out flows that are not completely established, i.e., those flows that do not complete the TCP hand-shake. These flows were mainly generated by scanning activity. We also filter out those flows that are not directed from internal hosts to external hosts. We further filter out flows whose destinations are well known as legitimate servers (e.g., Google, Youtube) that will unlikely be C&C servers of botnets. The white list is based on the top 1000 popular websites from Alexa.com.

In addition, we aggregate related flows into communication flows (C-flows) like [8]. The flows that share the same protocol, source IP, destination IP and port within an epoch are defined as a C-flow.

3.2. Similarity-based analysis

The purpose of similarity-based analysis is to cluster similar aggregated C-flows. The bots within the same botnet are controlled by the same botmaster. Thus, the communication patterns of flows generated by bots may exhibit more uniform behaviors than normal hosts in network traffic.

We summarize 15 features based on the traffic behaviors of bots, including duration, the number of packets within an flow and other statistical features. Table 1 describes these 15 statistical flow-based features in details, excluding the source and destination IP and port. To indicate the discrete sample distribution of each 15 statistical features, we first compute the quantiles[1] and divide the x-axis in 13 intervals as [0, k_1],...,[k_{12}, k_{13}], in which k_1 = q5%,...,k_{12} = q80%, k_{13} = q90%. For each characteristic of C-flows we can describe its variable distribution as 13 dimensional vectors, in which each element represents the number of flows within the corresponding interval.

Algorithm 1 shows the algorithm for calculating similarity score. θ is the scale coefficient for getting the best number of cluster centers. We first determine k, the number of clustering centers as the input for k-means. The k-means clustering is then executed. More specifically, we use K-means++ in order to avoid the algorithm getting into the local optimum. Finally, we calculate the average distance of each C-flow to the remaining C-flows within the same cluster, *avg*. The similarity score is define as 1-*avg*. The bigger the similarity score of C-flow is, the more abnormal the C-flow will be. The similarity score between C_i and C_j is defined as:

$$sim_{i,j} = \frac{C_i \cdot C_j}{\|C_i\| * \|C_j\|} \tag{1}$$

[1] The quantile ql% of a random variable X is the value q for which $P(X < q) = l$%.

Algorithm 1 Similarity-based detector of BotMark.

Input: The set of C-flows represented by features for clustering, $\{C_1, C_2, \ldots, C_n\}$;
Output: The similarity score of C-flows, $\{sim_1, sim_2, \ldots, sim_n\}$;
1: Select a C-flow as the first random cluster center, z_1;
2: Select the second cluster center z_2 that has maximum distance from z_1;
3: Add z_1, z_2 to the set of centers, $\{Z\}$;
4: **for** $i = 1 \rightarrow N$ **do**
5: **for** $j = 1 \rightarrow len(\mathrm{Z})$ **do**
6: $d_{ij} = \|C_i - z_j\|$;
7: **end for**
8: $d_i = \min(d_{i1}, \ldots, d_{ij})$;
9: **end for**
10: **if** $\max_i(d_1, \ldots, d_i) \geq \theta \ ||z_1 - z_2||$, $0 < \theta < 1$ **then**
11: $z = z \cup C_i$;
12: **goto** step 4;
13: **end if**// Get the cluster centers K
14: **repeat**
15: Add each C-flow into the nearest cluster center;
16: Recalculate the cluster center;
17: **until** Cluster center no longer changes or reach the maximum number of iterations;
18: **for** $p = 1 \rightarrow N$ **do**
19: Calculate the average distance between a C-flow and remaining C-flows within the same cluster, *avgd*;
20: S_{sim}=1-avgd;
21: **end for**

3.3. Stability-based analysis

The purpose of the stability-based analysis is to measure the stability of individual C-flow. The bots are detected based on the fact that the packets length distribution of botnets C-flows are relatively stable during its life time. In this work, we set a mutation threshold T_h to determine whether the C-flow is bot or not. If the distance between the two vectors representing the distributions of packet length in two adjacent sequences is less than T_h, it indicates that the C-flow has no mutation in the two sequences. In other words, the C-flow is more likely generated by the bots.

We use the following distance formula to measure the variation of the packet length distributions of C-flow in two adjacent time sequences. P_a and P_b represent the packet length distribution vector of an C-flow in the previous and the followed time sequences, respectively. The distance between P_a and P_b is defined as:

$$D(P_a, P_b) = \frac{1}{n} \sum_{k=1}^{w} \left(\frac{|P_a(k) - P_b(k)|}{1 + \max\{P_a(k), P_b(k)\}} \right) \tag{2}$$

where w is the dimensions packet size distribution vectors.

Algorithm 2 shows the algorithm for calculating stability score. The input parameters of algorithm include S and T_h. S is length of time sequences and T_h is the mutation threshold mentioned above.

3.4. Graph-based analysis

The purpose of the graph-based analysis is to first identify the C&C servers. We then mark the hosts connected with corresponding C&C servers as bots. In this work, we are motivated to mine the patterns that neighborhoods of normal nodes look like in traffic graph. We observe that the neighborhoods of normal nodes obey $y \propto x^{\alpha}$ power law distribution (patterns). As anomalous neighborhoods significantly differs from normal neighborhoods, we mark those nodes that deviate from the patterns as anomalous.

In this work, we focus on the individual egonet, rather than on the network. An egonet consists of a focal node and the nodes to whom ego is directly connected to plus the ties, if any, among the alters. Formally, an egonet is known as the induced 1-step neighborhood subgraph for a number of nodes in the traffic network. The huge number of features can be calculated from the egonet easily, such as the number of neighbors, the number of edges, total weights, principal eigenvalue, triplets of features. In this work, we use *N, E* and *W* as the features, in which *N* and *E* are paired and *N* is denoted as the number of neighbors, *E* as the number of edges, and *W* as the total weight in an egonet.

Note that in the power law we have $y = Cx^{\alpha}$ which means that $\log(y) = \log(C) + \alpha \log(x)$. We use least-square technique to fit a straight line to measure the anomalies of an ego. The anomaly score of ego i is defined as:

$$out\_line(i) = \frac{\max(E_i, CN_i^{\alpha})}{\min(E_i, CN_i^{\alpha})} * \log(E_i - CN_i^{\alpha} + 1) \tag{3}$$

Algorithm 2 Stability-based detector of BotMark.

Input: The set of all packets length of within an C-flow C_k, T_h, S;
Output: The stability score of C-flow, $Ssta_k$;
Compute q%5, q%10, ..., q%90 of all packets length, Obtain $[k_1, k_2, ..., k_{13}]$;
Compute the number of time sequences. N = The difference between arrival time of the first and last packet /S;
for $i = 1 \rightarrow N$ **do**
Approximate distribution of packets within i sequence p_i;
end for ▷ //Obtain $\{p_1, p_2,...,p_N\}$
for $i = 1 \rightarrow N$ **do**
mc=0; // the counts of mutation
$j \leftarrow i + 1$;
Calculate $D(P_i, P_j)$;
if $D(P_i, P_j) > T_h$ **then**
$mc \leftarrow mc + 1$;
end if
$Ssta_i = 1 - \frac{mc}{S-1}$;
end for

where N_i, E_i denote the value of feature pair of ego i, respectively. It is clear that an ego will be identified as an anomaly if it does not obey the power law, i.e., $out\_line(i)$ is bigger than a threshold.

However, least-square method may yield false positives as distributed software hosts generate similar traffic behaviours with C&C servers, such as Hadoop, Zookeeper, Redis. We use Local Outlier Factor (LOF) to detect anomalies (or outliers) to help in detecting anomalies. In addition, we observe that the infected hosts within the same botnet make connection with C&C server frequently to receive and execute commands. We also introduce the Coefficient of Variance (CV) to the anomaly scores. The CV is defined as σ/μ, where σ and μ represent the standard deviation and the average W respectively. We sum the three terms, namely, $out\_line(i)$, $LOF(i)$ and $CV(i)$ to obtain the final anomaly scores of an ego.

3.5. Hybrid analysis

The BotMark overcomes the weaknesses and limitation of previous C-flow based features or graph-based features of network traffic. It performs automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors by ensemble of the detection results based on similarity scores, stability scores and anomaly scores and make a final decision. More specifically, if more than two detectors identify the specific host as a bot, the host is then marked as a bot. Otherwise, it will be marked as normal.

4. Evaluation

4.1. Data sets

We have collected a very large size of network traffic by simulating 5 newly propagated botnets in a real computing environment, including Mirai, Zues, Athena, Black energy and Ares. We share our data in the research community.[2]

The botnet data sets used in this work consist of TCP traffic starting at 18:47:09 on August 4, 2016. In these botnet traffic traces, the traffic of Mirai that was first found in 2016 was crawled for 16 days and its size is 5.1Gb. Table 3 lists the basic information about these botnet traffic traces. Among them, Mirai is Telnet-based botnet, and the others are HTTP-based botnets.

The background traffic is collected from a Internet Service Providers (ISP). There are about 200,000 hosts in background traffic. We have carefully labeled the botnet datasets. The total number of packets is 456,229,550 in our experiments, 455,586,270 (about 99.86%) of which are normal traffic data packets. We convert data packets into C-flows as described in Section 3.1. Finally we obtain 305 anomalous botnets C-flows as well as 7,183,379 normal C-flows. The datasets and their statistics of network traffic data are shown in Table 2.

4.2. Experimental results analysis

The experiments were run on a 10-core Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz with 330 GB main memory with OS as Ubuntu 16.04.4 LTS (GNU/Linux 4.13.0-45-generic x86_64). The performance of BotMark is evaluated with F-score. F-score is defined as

$$F - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

[2] **Data Sets available at**: http://infosec.bjtu.edu.cn/wangwei/page_id=85.

Table 2
Amount of data on each botnet trace.

Trace	# Pkts	# Flows	# botnet Flows	# C&C Flows	# C-flows	# Botnet C-Flows	# C&C C-Flows
Mirai	96,558(0.0212%)	160	80	80	160	80	80
Zeus	59,748(0.0131%)	6458	3229	3229	3309	80	3229
Ares	81,217(0.0178%)	23,629	16,443	7186	7266	80	7186
Athena	18,971(0.0042%)	3654	1827	1827	1832	5	1827
Black energy	386,786(0.0848%)	73,366	35,183	35,183	35,243	60	35,183

Table 3
Basic information about five different types of botnets.

Traces	Duration	Size	# Bots	C&C	Notes
Mirai	380h	5.1G	80	1	DDoS attacks
Athena	90h	3.2G	5	1	DDoS attacks
Blackenergy	140h	0.99G	60	1	Spaming
Zeus	123h	1.2G	80	1	Steal banking information
Ares	63h	2.04G	80	1	Keylogger, File download, Screen monitor

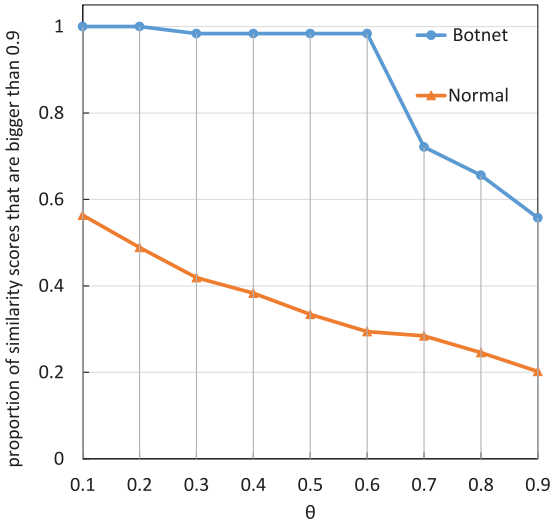

Fig. 2. The effect of θ on the C-flow similarity score.

where Precision is the proportion of True Positive (TP) to all the positive results, and Recall is also called True Positive Rate (TPR) defined as the proportion of TP in all the positive instances.

In the following subsections, we first evaluate the effectiveness of similarity-based detector and stability-based detector, respectively. Second we combine the two detectors and discuss its performance. Third, we evaluate the effectiveness of graph detector. Finally, we analyze and discuss the performance with the hybrid analysis of three detectors, namely, similarity-based detector, stability-based detector and graph detector based on the network traffic.

4.2.1. Similarity-based analysis

We analyze the C-flows whose similarity scores are bigger than 0.9 in order to set the scale coefficient θ. It can also be set as another value that is close to 1, as the similarity scores among all the malicious C-flows are very similar and close to 1. From Fig. 2, it is observed that when the value of θ is small, e.g., 0.1, the number of clusters is large and thus the clusters formed by normal C-flows are also relatively tight, and this would result in lots of false positives. On the contrary, if θ is big, there are much more C-flows within a cluster due to the small number of clusters. Thus, the similarity-based detector may not distinguish malicious C-flows from normal C-flows, resulting in false negatives. In order to make sure that the similarity scores of botnet C-flows are as big as possible while the similarity scores of normal C-flows are as small as possible, the values of θ in {0.4, 0.5, 0.6} are all acceptable from Fig. 2. After setting θ, we perform the k-means clustering and then calculate the similarity score. Since the performance of similarity analysis heavily depends on value of θ, detailed results are presented in Fig. 3 in terms of Receiver Operating Characteristic (ROC) curves that show the relation between true positive rates and false positive rates, we finally assign $\theta = 0.6$ in our experiment.

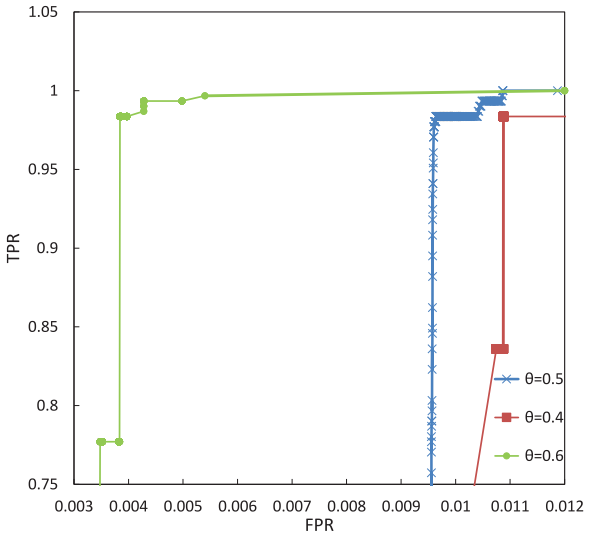

Fig. 3. ROC curves with the similarity-based detector.

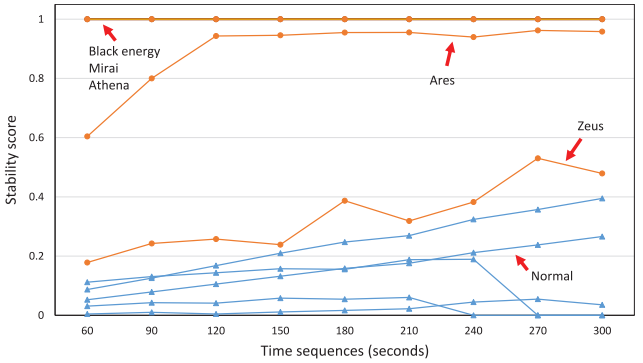

Fig. 4. The effect of time sequences.

4.2.2. Stability-based analysis

In order to prove that the packet size distribution generated by the bots is more stable than the distribution of normal hosts. Intuitively, We randomly select five normal C-flows (blue line) as well as five representative bot C-flows (red line). Fig. 4 illustrates the effect of time sequences on stability scores. From the Figure, it is observed that (1) the stability scores of the botnet C-flows are significantly bigger than those of the normal C-flows; (2) as the size of sequence increases, the number of time sequences decreases during the detection time, thus the stability score increases; (3) some C-flows may be communication suspension, finally resulting in the zero of stability score. In order to distinguish botnet C-flows from normal C-flows, we set the size of sequences as 180 s in this work.

We also analyze the effect of mutation threshold (T_h) on the stability scores. As shown in Fig. 5, the ROC curves with different T_h is clearly displayed. It is seen that the performance outperforms the others when mutation threshold is given 0.3. That is, If the distance between the two distributions of packet length is less than T_h in two adjacent sequences, it indicates that the C-flow has no mutation. In other words, the C-flow is more likely generated by the bots.

4.2.3. Flow-based analysis

Table 4 shows the detection performance with the combination of the similarity-based and stability-based detector with different type of botnets. It is observed that the similarity-based detection can identify the Zeus botnet that are unable to be identified by the stability-based detector. The stability-based detector accurately detects the Athena botnet with 100% in terms of TPR, while the similarity-based detector do not. We thus make hybrid analysis based on the combination of similarity-based detector and stability-based detector to make up for the deficiency of individual detectors. S_{sim} and S_{sta} represent the anomaly scores generated by similarity-based detector and by stability-based detector. The hybrid detection results S_{C-flow} are defined as

$$S_{C-flow} = \alpha S_{sim} + \beta S_{sta} \tag{5}$$

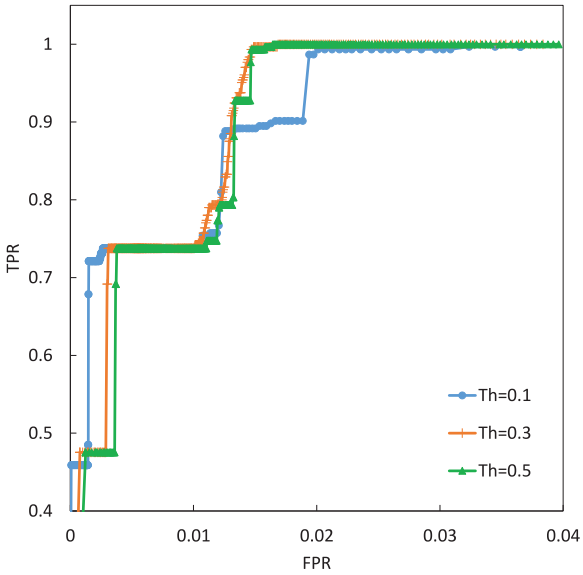

Fig. 5. ROC curves with stability-based detector.

Table 4
Detection Rates with traditional detectors.

Botnets	Similarity	Stability	C-flow
Mirai	100%	100%	100%
Ares	100%	100%	100%
Black-energy	100%	100%	100%
Zeus	100%	66.25%	93.75%
Athena	0	100%	100%
All	98.36%	91.15%	98.36%

Table 5
The performance with traditional detectors.

Method	*F*-score	Acc	TPR	FPR
Similarity	0.087247	0.9904	0.9836	0.009645
Stability	0.060732	0.9868	0.9115	0.013181
C-flow	0.152788	0.9949	0.9836	0.005108

where α and β represent the weights of similarity-based and of stability-based detectors. In this work, we adjust the weights of the similarity-based detector from 0.1 to 1, and empirically assign $\alpha = 0.6$ and $\beta = 0.4$. In addition, we compare the performance generated with similarity-based detector, stability-based detector and hybrid detector. From Table 5, it is observed that the hybrid analysis has no obvious effect in terms of detection rates. It is worth mentioning that the false positive rates decrease from 0.96% to 0.51% after hybrid analysis by combining the similarity-based detector and stability-based detector. In general, the hybrid analysis outperforms any individual detector in terms of accuracy with 99.49%.

4.2.4. Graph-based analysis

To get a better fit, we bin the data into exponentially wider buckets (they will appear evenly spaced on a log scale). Fig. 6 shows the *E* versus *N* for all egonets with log-log scales. The red line is the least squares fit. The black circles represent median values for each buckets. The blue circles are median values that corrects the fit line with Cook distance. Red triangles are considered as anomalies, or C&C servers. The blue dashed and the black dashed lines represent the value boundaries of all the data. The top 5 anomalies are marked to identify the 5 types of botnets' C&C servers. Once we identify the 5 C&C servers, all the hosts connected with corresponding servers are accurately marked as bots. We further analyze the detection performance of graph-based detector with different types of botnets in Table 6. We rank the anomaly scores of all the egos and then mark the top five nodes as C&C servers. Unfortunately, the Athena is not ranked on the top five. Instead, the total number of hosts that are connected to the fifth toppest server are 16926, thus results in 16,926 false positives. As a result, the graph-based detector achieves detection accuracy as 91.66%.

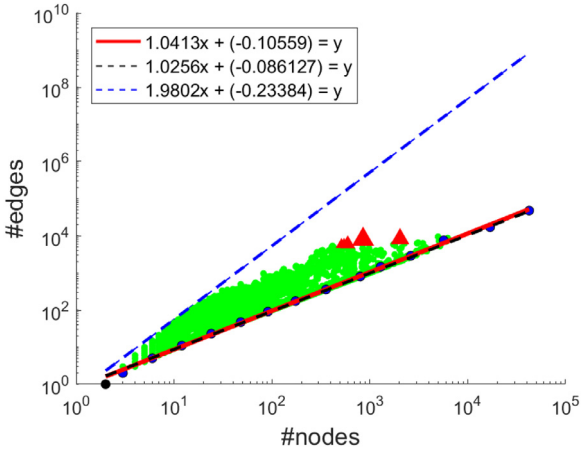

Fig. 6. Depiction of the power law followed by edges and nodes.

Table 6
Detection performance with graph-based detector.

Botnets	Rank	#bots	Detected?	Detection Rate
Mirai	1	80	YES	100%
Ares	2	80	YES	100%
Black-energy	3	60	YES	100%
Zeus	4	80	YES	100%
Athena	234	5	NO	0

Table 7
Detection performance with hybrid analysis.

Method	*F*-score	Acc	TPR	FPR
Similarity	0.087247	0.9904	0.9836	0.009645
Stability	0.060732	0.9868	0.9115	0.013181
Graph	0.034811	0.9166	0.9836	0.083478
BotMark	0.115207	0.9994	0.9836	0.000641

4.2.5. Botmark

We make the hybrid analysis based on flow-based and graph-based traffic behaviors by voting. Extensive experimental results demonstrate the effectiveness of BotMark. As the number of false positives is considerably reduced by voting on the results generated by three types of detectors, namely similarity-based detector, stability-based detector and graph-based detector, the detection accuracy is certainly improved. As shown in Table 7, the flow-based detector achieves the detection accuracy of 99.49%, while the graph-based detector achieves 91.66%. BotMark achieves the 99.94% in terms of detection accuracy with hybrid analysis. The reason the similarity-based detector does not work on Athena is that the C-flows generated by Athena are tightly clustered with normal C-flows. Moreover, the C-flows generated by Zeus are relatively unstable during its lifetime due to their characteristics. The stability-based detector thus marks some C-flows of Zeus as normal hosts easily. The graph-based detector also cannot detect its C&C server at all. As a consequence, BotMark detects bots with hybrid analysis based on flow-based and graph-based traffic behaviors. It outperforms any individual detector with suppressing the false positive rates and improving the detection accuracy. The experimental results also show that flow-based detector are more effective for botnets detection than graph-based detector.

5. Limitations

Botnets may try to utilize a legitimate server as their C&C communication to evading the detection. If we whitelist such legitimate servers to reduce the volume of traffic and improve the efficiency of BotMark, it would be immediately invalid. Other advanced evasion techniques may be attempted to mimic the communication patterns of normal hosts, i.e., randomizing the number of packets per flow and the number of bytes per packet to evade the detection. The flow based detector would be thus immediately invalid. In addition, some distributed softwares generate similar traffic behaviour with legitimate C&C servers to make sure their slaves alive, such as Redis, Zookeeper, and this would result in false negatives.

6. Conclusion

The fact that bots have evolved continuously and become increasingly sophisticated calls for more effective detection models. In this work, we propose BotMark that automatically detects bots with hybrid analysis of flow-based and graph-based traffic behaviors. The flow-based detector consists of similarity-based component and stability-based component. The hybrid of both detectors can characterize the botnet behaviors more comprehensively than any individual detector. Moreover, Botmark is independent of botnet C&C protocols and structures, requires no *a priori* knowledge of botnets, and thus can be adopted in complex networking environments. We employ machine learning algorithms including *k*-means clustering, least-square technique and Local Outlier Factor (LOF) for building detection models. Extensive experimental results demonstrate the effectiveness of BotMark. It achieves the detection accuracy of 99.94% with hybrid analysis of flow-based and graph-based traffic behaviors, outperforming any individual detector. The experimental results also demonstrate the effectiveness of BotMark on the detection of novel botnets like Mirai, Athena and Black energy. In future work, we plan to explore more effective graph-based features to better characterize bots so as to improve the detection results.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The work reported in this paper was supported in part by Natural Science Foundation of China, under Grant U1736114, and in part by CNCERT/CC, under Grant number K19GY500020.

References

[1] L. Akoglu, M. McGlohon, C. Faloutsos, OddBall: spotting anomalies in weighted graphs, in: Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21–24, 2010. Proceedings. Part II, 2010, pp. 410–421.
[2] L. Akoglu, H. Tong, D. Koutra, Graph based anomaly detection and description: a survey, Data Min. Knowl. Discov. 29 (3) (2015) 626–688.
[3] F.V. Alejandre, N.C. Cortés, E.A. Anaya, Feature selection to detect botnets using machine learning algorithms, in: 2017 International Conference on Electronics, Communications and Computers, CONIELECOMP 2017, Cholula, Mexico, February 22–24, 2017, 2017, pp. 1–7.
[4] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech 2008 (10) (2008) P10008.
[5] H. Choi, H. Lee, H. Lee, H. Kim, Botnet detection by monitoring group activities in DNS traffic, in: Seventh International Conference on Computer and Information Technology (CIT 2007), October 16–19, 2007, University of Aizu, Fukushima, Japan, 2007, pp. 715–720.
[6] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H.R. Medal, M. Marufuzzaman, L. Bian, Botnet detection using graph-based feature clustering, J. Big Data 4 (2017) 14.
[7] J. François, S. Wang, T. Engel, et al., Bottrack: tracking botnets using netflow and pagerank, in: International Conference on Research in Networking, Springer, 2011, pp. 1–14.
[8] G. Gu, R. Perdisci, J. Zhang, W. Lee, BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection, in: Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA, 2008, pp. 139–154.
[9] R. Hassanzadeh, R. Nayak, D. Stebila, Analyzing the effectiveness of graph metrics for anomaly detection in online social networks, in: Web Information Systems Engineering - WISE 2012 - 13th International Conference, Paphos, Cyprus, November 28–30, 2012. Proceedings, 2012, pp. 624–630.
[10] M. Iliofotou, H. chul Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, G. Varghese, Graption: a graph-based P2P traffic classification framework for the internet backbone, Comput. Netw. 55 (8) (2011) 1909–1920.
[11] N. Kheir, C. Wolley, BotSuer: suing stealthy P2P bots in network traffic through netflow analysis, in: Cryptology and Network Security - 12th International Conference, CANS 2013, Paraty, Brazil, November 20–22. 2013. Proceedings, 2013, pp. 162–178.
[12] G. Kirubavathi, R. Anitha, Botnet detection via mining of traffic flow characteristics, Comput. Electr. Eng. 50 (2016) 91–101.
[13] S. Lagraa, J. François, A. Lahmadi, M. Miner, C.A. Hammerschmidt, R. State, Botgm: unsupervised graph mining to detect botnets in traffic flows, in: 1st Cyber Security in Networking Conference, CSNet 2017, Rio de Janeiro, Brazil, October 18–20, 2017, 2017, pp. 1–8.
[14] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, Z. Zhang, CreditCoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles, IEEE Trans. Intell. Transp. Syst. 19 (7) (2018) 2204–2220, doi:10.1109/TITS.2017.2777990.
[15] W.H. Liao, C.C. Chang, Peer to peer botnet detection using data mining scheme, in: 2010 International Conference on Internet Technology and Applications, 2010, pp. 1–4.
[16] X. Liu, J. Liu, S. Zhu, W. Wang, X. Zhang, Privacy risk analysis and mitigation of analytics libraries in the android ecosystem, IEEE Trans. Mob. Comput. (2019), doi:10.1109/TMC.2019.2903186.
[17] C. Livadas, R. Walsh, D. Lapsley, W.T. Strayer, Usilng machine learning technliques to identify botnet traffic, in: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 2006, pp. 967–974.
[18] S. Nagaraja, P. Mittal, C. Hong, M. Caesar, N. Borisov, BotGrep: finding P2P bots with structured graph analysis, in: 19th USENIX Security Symposium, Washington, DC, USA, August 11–13, 2010, Proceedings, 2010, pp. 95–110.
[19] R.S. Rawat, E.S. Pilli, R.C. Joshi, Survey of peer-to-peer botnets and detection frameworks, I. J. Netw. Secur. 20 (3) (2018) 547–557.
[20] S. Saad, I. Traoré, A.A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, P. Hakimian, Detecting P2P botnets through network behavior analysis and machine learning, in: Ninth Annual Conference on Privacy, Security and Trust, PST 2011, 19–21 July, 2011, Montreal, Québec, Canada, 2011, pp. 174–180.
[21] E.B.B. Samani, H.H. Jazi, N. Stakhanova, A.A. Ghorbani, Towards effective feature selection in machine learning-based botnet detection approaches, in: IEEE Conference on Communications and Network Security, CNS 2014, San Francisco, CA, USA, October 29–31, 2014, 2014, pp. 247–255.
[22] Y. Shang, S. Yang, W. Wang, Botnet detection with hybrid analysis on flow based and graph based features of network traffic, in: Cloud Computing and Security. Lecture Notes in Computer Science, vol 11064. Springer, Cham., Springer International Publishing, 2018, pp. 612–621.
[23] K. Singh, S.C. Guntuku, A. Thakur, C. Hota, Big data analytics framework for peer-to-peer botnet detection using random forests, Inf. Sci. 278 (2014) 488–497.
[24] F. Tegeler, X. Fu, G. Vigna, C. Kruegel, Botfinder: finding bots in network traffic without deep packet inspection, in: Conference on emerging Networking Experiments and Technologies, CoNEXT '12, Nice, France - December 10, – 13, 2012, 2012, pp. 349–360.
[25] J. Wang, I.C. Paschalidis, Botnet detection using social graph analysis, in: 52nd Annual Allerton Conference on Communication, Control, and Computing, Allerton 2014, Allerton Park & Retreat Center, Monticello, IL, September 30, – October 3, 2014, 2014, pp. 393–400.

[26] W. Wang, B. Fang, Z. Zhang, C. Li, A novel approach to detect IRC-based botnets, in: 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, vol. 1, 2009, pp. 408–411.
[27] W. Wang, X. Guan, X. Zhang, Processing of massive audit data streams for real-time anomaly intrusion detection, Comput. Commun. 31 (1) (2008) 58–72.
[28] W. Wang, X. Guan, X. Zhang, L. Yang, Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data, Comput. Secur. 25 (7) (2006) 539–550.
[29] W. Wang, T. Guyet, R. Quiniou, M. Cordier, F. Masseglia, X. Zhang, Autonomic intrusion detection: adaptively detecting anomalies over unlabeled audit data streams in computer networks, Knowl.-Based Syst. 70 (2014) 103–117.
[30] W. Wang, Y. He, J. Liu, S. Gombault, Constructing important features from massive network traffic for lightweight intrusion detection, IET Inf. Secur. 9 (6) (2015) 374–379.
[31] W. Wang, Y. Li, X. Wang, J. Liu, X. Zhang, Detecting android malicious apps and categorizing benign apps with ensemble of classifiers, Future Gener. Comp. Syst. 78 (2018) 987–994.
[32] W. Wang, J. Liu, G. Pitsilis, X. Zhang, Abstracting massive data for lightweight intrusion detection in computer networks, Inf. Sci. 433–434 (2018) 417–430.
[33] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han, X. Zhang, Exploring permission-induced risk in android applications for malicious application detection, IEEE Trans. Inf. Forensics Secur. 9 (11) (2014) 1869–1882.
[34] W. Wang, Y. Wang, X. Tan, Y. Liu, S. Yang, BotCapturer: detecting botnets based on two-layered analysis with graph anomaly detection and network traffic clustering, Int. J. Performabil.Eng. 14 (5) (2018) pp.1050–1059.
[35] W. Wang, X. Zhang, S. Gombault, Constructing attribute weights from computer audit data for effective intrusion detection, J. Syst. Softw. 82 (12) (2009) 1974–1981.
[36] W. Wang, M. Zhao, Z. Gao, G. Xu, H. Xian, Y. Li, X. Zhang, Constructing features for detecting android malicious applications: issues, taxonomy and directions, IEEE Access 7 (2019) 67602–67631, doi:10.1109/ACCESS.2019.2918139.
[37] W. Wang, M. Zhao, J. Wang, Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network, J. Ambient Intell. Hum. Comput. 10 (8) (2019) 3035–3043.
[38] X. Wang, W. Wang, Y. He, J. Liu, Z. Han, X. Zhang, Characterizing Android apps behavior for effective detection of malapps at large scale, Future Gener. Comput. Syst. 75 (2017).
[39] X. Xu, N. Yuruk, Z. Feng, T.A.J. Schweiger, SCAN: a structural clustering algorithm for networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12–15, 2007, 2007, pp. 824–833.
[40] X. Yu, X. Dong, G. Yu, Y. Qin, D. Yue, Data-adaptive clustering analysis for online botnet detection, in: 2010 Third International Joint Conference on Computational Science and Optimization, vol. 1, 2010, pp. 456–460.
[41] D. Zhao, I. Traoré, B. Sayed, W. Lu, S. Saad, A.A. Ghorbani, D. Garant, Botnet detection based on traffic behavior analysis and flow intervals, Comput. Secur. 39 (2013) 2–16.

Wei Wang is currently a full professor in the Department of Information Security, Beijing Jiaotong University, China. He earned his Ph.D. degree in control science and engineering from Xi'an Jiaotong University, in 2006. He was a postdoctoral researcher in University of Trento, Italy, during 2005–2006. He was a postdoctoral researcher in TELECOM Bretagne and in INRIA, France, during 2007–2008. He was a European ERCIM Fellow in Norwegian University of Science and Technology (NTNU), Norway, and in Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, during 2009–2011. He has authored or co-authored over 80 peer-reviewed papers in various journals and international conferences. He is an Editorial Board member of Computers & Security and a Young AE of Frontiers of Computer Science. His main research interests include mobile, computer and network security.

Yaoyao Shang is currently a M.S. student in the School of Computer and Information Technology, Beijing Jiaotong University, China. She received her B.S. degree from Shanxi University, China, in 2016. Her main research interests lie in anomaly detection.

Yongzhong He is currently an associate professor with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. He earned his Ph.D. degree from Graduate School of Chinese Academy of Sciences in 2006. He has authored or coauthored over 30 peer-reviewed papers in journals and conferences. His main research interests include system security and privacy.

Yidong Li is the Vice-Dean and a professor in the School of Computer and Information Technology at Beijing Jiaotong University. Dr. Li received his B.Eng. degree in electrical and electronic engineering from Beijing Jiaotong University in 2003, and M.Sci. and Ph.D. degrees in computer science from the University of Adelaide, in 2006 and 2010, respectively. Dr. Li's research interests include big data analysis, privacy preserving and information security, data mining, social computing and intelligent transportation. Dr. Li has published over 80 research papers in various journals (such as IEEE Trans. on Information Forensics & Security, IEEE Trans. on Intelligent Transportation Systems), and refereed conferences. He has also co-authored/co-edited 5 books (including proceedings) and contributed several book chapters. He has organized several international conferences and workshops and has also served as a program committee member for several major international conferences such as PAKDD, NFOSCALE, WAC, SAC, PDCAT, DANTH, and PAAP.

Jiqiang Liu received his B.S. (1994) and Ph.D. (1999) degree from Beijing Normal University. He is currently a Professor at the School of Computer and Information Technology, Beijing Jiaotong University. He has published over 70 scientific papers in various journals and international conferences. His main research interests are trusted computing, cryptographic protocols, privacy preserving and network security.