Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4916953, 9 pages
https://doi.org/10.1155/2017/4916953

Research Article

Identifying APT Malware Domain Based on Mobile DNS Logging

Weina Niu,[1,2] Xiaosong Zhang,[1,2] GuoWu Yang,[2] Jianan Zhu,[3] and Zhongwei Ren[1]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*
[2] *Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*
[3] *School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China*

Correspondence should be addressed to Xiaosong Zhang; johnsonzxs@uestc.edu.cn

Received 25 January 2017; Accepted 7 March 2017; Published 6 April 2017

Academic Editor: Lixiang Li

Advanced Persistent Threat (APT) is a serious threat against sensitive information. Current detection approaches are time-consuming since they detect APT attack by in-depth analysis of massive amounts of data after data breaches. Specifically, APT attackers make use of DNS to locate their command and control (C&C) servers and victims' machines. In this paper, we propose an efficient approach to detect APT malware C&C domain with high accuracy by analyzing DNS logs. We first extract 15 features from DNS logs of mobile devices. According to Alexa ranking and the VirusTotal's judgement result, we give each domain a score. Then, we select the most normal domains by the score metric. Finally, we utilize our anomaly detection algorithm, called Global Abnormal Forest (GAF), to identify malware C&C domains. We conduct a performance analysis to demonstrate that our approach is more efficient than other existing works in terms of calculation efficiency and recognition accuracy. Compared with Local Outlier Factor (LOF), k-Nearest Neighbor (KNN), and Isolation Forest (iForest), our approach obtains more than 99% F-M and R for the detection of C&C domains. Our approach not only can reduce data volume that needs to be recorded and analyzed but also can be applicable to unsupervised learning.

1. Introduction

Advanced Persistent Threat (APT) [1, 2] is an attack that is launched by the well-funded and skilled organization to steal high-value information for a long time. APT attackers would install malware on the compromised machine to build command and control (C&C) channel after infiltrating into the targeted network. Most malware makes use of Domain Name System (DNS) to locate their domain name servers and compromised devices. Then, APT attackers can establish long-term connection to victims' devices for stealing sensitive data. Thus, malware C&C domain detection can help security analysts to block essential stage of APT.

Currently, there are some works to identify C&C domain by analyzing network traffic about PC [3–8]. BotSniffer [3], BotGAD [4], and BotMiner [5] made use of specific behavior anomaly (e.g., daily similarity and short life) to detect C&C involved in a botnet. The main reason is that bot hosts have group similarity. Other works [6–8] also distinguish between malicious domains and normal domains according to domain-based features, such as domain name string composition, registration time, and active time. However, these detection approaches cannot be applied to APT malware since APT attackers infect a small number of machines, and they behave normally to avoid detection. Machine learning technology is proved to be effective in identifying malware [6]. However, there are few artificially marked data of APT malware. Moreover, normal and abnormal samples overlap with each other.

In order to address these challenges, we propose an approach to identifying APT malware domains based on DNS logs. We conduct experiments to evaluate our proposed algorithm, called Global Abnormal Forest (GAF), with three traditional algorithms, namely, Local Outlier Factor (LOF),

k-Nearest Neighbor combined with LOF (LOF-KNN), and Isolation Forest (iForest). The experimental results demonstrate that our proposed algorithm behaves best on a dataset consisting of 300000 DNS requests each day from a regional base station. Specifically, the contributions of this work are specified as follows:

(i) We characterize statistics of normal domains and define a rule based on Alexa and VirusTotal to select the most normal domains.

(ii) We extract 15 features of mobile DNS requests in multigranularity by studying large DNS logs in a real dynamic network environment consisting of 10K devices with more than 300,000 DNS requests per day.

(iii) We propose an anomaly detection algorithm to compromise accuracy and efficiency of C&C domains detection by introducing differentiated information entropy.

The structure of this paper is arranged as follows. we motivate the need for APT malware C&C detection using anomaly detection in Section 2; Section 3 presents an overview of the proposed approach and introduces the most normal domain identification rules, and we motivate the choice for features that are related to APT malware C&C domain in Section 3; Section 4 describes the building of our anomaly detection model; Section 5 completes experimental evaluation metrics and illustrates the experimental results of different algorithms; Section 6 introduces the related work; Section 7 makes a conclusion of the paper.

2. Background on C&C Detection Using Anomaly Detection

APT was first used in 2006 and has become widely known since the exposure of Google Aurora in 2010 [7]. In 2013, the APT attack was pushed to cusp due to PRISM. Thus, the APT attack has brought new challenges to cybersecurity due to long-latent, intelligence penetration and overcustomization [8, 9]. APT attackers often install DNS-based APT malware, for instance, Trojan horse or backdoor, on the infected machine for stealing sensitive data and hiding the real attack source. Identifying malware during their command control channel establishment phase is a good choice. However, DNS behavioral features of compromised machines infected by APT malware are different from the botnet. Thus, APT malware identification based on DNS data is a challenge.

Suspicious instances of APT malware are rare and the amount of data cannot be fully labeled by the expert. The most normal domain instances within the DNS data are available. Moreover, anomaly detection [10] can identify new and unknown attack since it does not depend on fixed signatures. Thus, we use anomaly detection to identify malware C&C domain using mobile DNS logs. The most common anomaly detection includes statistical anomaly detection, classification-based anomaly detection, and clustering-based anomaly detection [11]. If the labeled set has been collected, classification-based anomaly detection, like Genetic Algorithm [12], Support Vector Machine [13], and Neural Network [14], is preferable. However, in the real APT attack, the label of data is very difficult to obtain. The unsupervised method can be used to identify malware C&C domain, such as LOF, LOF-KNN, and iForest. LOF [15] determines whether the data is an outlier according to neighbor density. LOF-KNN [16] identifies outlier according to similarity. However, these two approaches have high computational complexity and too many false alarms. To ease these two problems, iForest [17] detects anomalies using the average path length of trees that requires a small subsampling size to achieve high detection performance. Thus, we can build partial models and exploit subsampling to identify malware C&C domain. Isolation Forest is based on the assumption that each instance is isolated to an external node when a tree is grown. Unfortunately, attribute values of normal domain and malware domain are relatively close. Moreover, traditional anomaly detection algorithms ignore the different influences of different properties. In this work, we introduce differentiated information entropy to improve the efficiency and utilize distance measures to detect anomalies.

3. Overview of Our Approach

In this section, we present an overview of the proposed approach for identifying APT malware domain, explain why we select those features that may be indicative of APT malware domain, and illustrate the metric for selecting the most normal domains.

3.1. Architecture of Our Approach. DNS logs are small but important. Thus, this work mainly focuses on the analysis of DNS logs in order to detect suspicious domains involved in APT malware. We store DNS logs that contain accessing user, source IP, destination IP, country flag, domain name, request time, and response time. Then we extract features according to logs and make use of anomaly detection technology to identify APT malware C&C domain. Figure 1 gives an overview of the system architecture of the proposed approach. The system consists of components including the following: (1) DNS logs collector stores the DNS logs produced by mobile devices in the network that is being monitored; (2) multigranularity feature extractor is responsible for extracting features of domains that are stored in DNS log database; (3) normal domain identifier is used to select the most normal domains; (4) anomaly learning module trains anomaly detector using malware domain that is labeled by experts from grey set and APT malware C&C domain produced by detector, normal instance from normal set; (5) anomaly detector takes decisions according to the identification results produced by the anomaly detection model.

The deployment of the system consists of three steps. In the first step, the features that we interested are extracted. Details and motivations on the chosen features will be discussed in Section 3.2. The second step defines a metric to select normal domain used to train. The third step involves

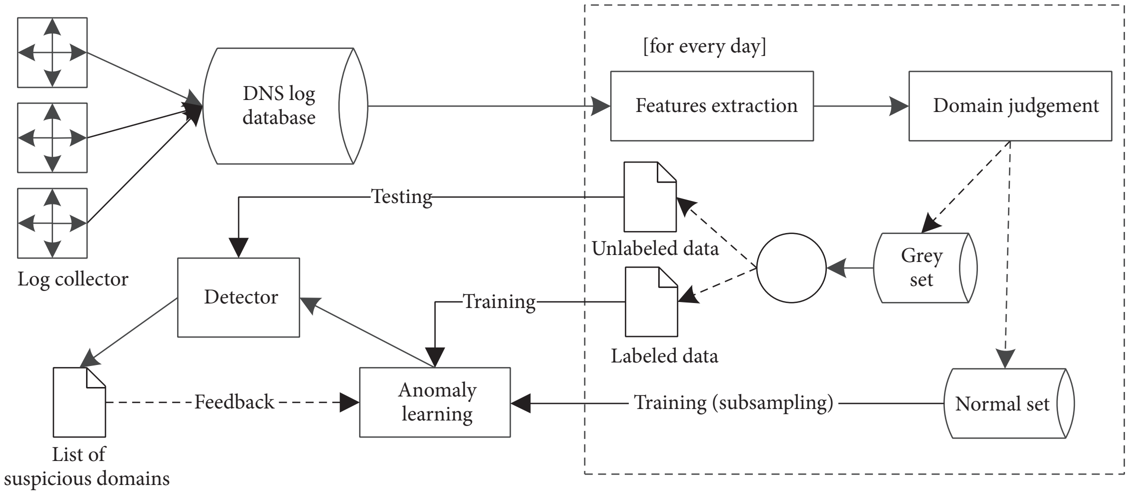


Figure 1: Framework of our proposed identification approach.

Table 1: Features of domain name.

FeatureSet	FeatureName
DNS request and answer-based features	Number of distinct source IP addresses Number of distinct IP addresses with the same domain IP in the same country using the predefined IP addresses
Domain-based features	Alexa ranking The length of domain The level of domain containing IP address
Time-based features	Request frequency Reaction time repeating pattern
whois-based features	Registration duration Active duration Update duration Number of DNS

our proposed anomaly detection algorithm, which uses part of normal samples to predict C&C domains. The proposed algorithm is described in detail in Section 4. The result is a list of the suspicious domains involved in APT malware.

3.2. Feature Extraction. In this work, we extracted 15 features to detect APT malware C&C domains based on mobile DNS logs. We also gave explanations of the 15 features and explained the reasons that they can be used to detect malicious domain. The extracted domain features are shown in Table 1.

3.2.1. DNS Request and Answer-Based Features. APT attackers usually use servers residing in different countries to build C&C channel in order to evade detection. Moreover, attackers make use of fast flux to hide the true attack source [18]. APT attacker changes the C&C domain to point to predefined IP addresses, such as look back address and invalid IP address. With this insight, we extracted three features from DNS request and response, such as the number of distinct source IP addresses, the number of distinct IP addresses with the same domain, IP in the same country, and using the predefined IP addresses.

3.2.2. Domain-Based Features. Attackers prefer to use the long domain to hide the doubtful part [19]. By analyzing the network traffic produced during the malware communicates with command and control servers, we find that many malware C&C domains have the following characteristics: high level, long string, containing IP address, and low visitor number. Thus, Alexa ranking, the length of the domain, the level of domain, and containing IP address are helpful in identifying malware domain. For example, if a domain name contains an IP address, such as "192.168.1.173.baidu.com", we would conclude that it may be a malicious domain.

3.2.3. Time-Based Features. When there is a connecting failure in the process of compromised device connect to the C&C server, compromised machine may send many repeated DNS requests. Sometimes, behaviors of these infected devices show similarities. Since IP address of malware domain is not stored in the local server, the domain name resolution takes longer time. Moreover, we observe that few domains have high query frequency through analyzing the domain access records during one day in our experimental environment, which is illustrated in Figure 2. This phenomenon helps us to further identify malicious domain names. Thus, we extracted three features to identify APT malware C&C domain, such as request frequency, reaction time, and repeating pattern.

3.2.4. Whois-Based Features. Trustworthy domains are regularly paid for several years in advance and they have a long

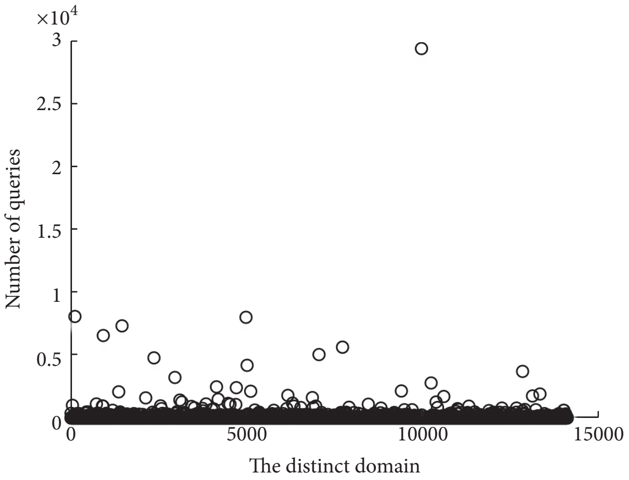


Figure 2: Distribution of query frequency of distinct domain.

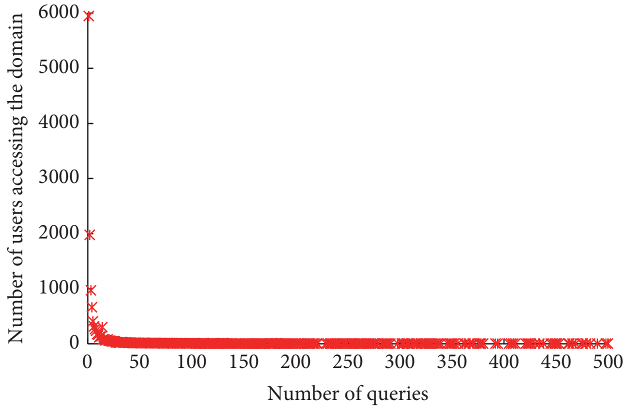


Figure 3: Distribution of the number of domain queries initiated by internal devices.

time to live [20]. However, most malware domains live for a short period of time, which is less than 6 months. Moreover, DNS record of the suspicious domain is empty or not found. Based on the above observation, we can use registration duration, active duration, update duration, and DNS record to detect malicious domain.

3.3. A Metric for Normal Domain Judgement. In order to implement anomaly detection, it is necessary to determine normal samples. An intuitive approach for selecting normal domains according to the number of DNS requests initiated by internal devices. However, in order to reduce exposure risk, APT attackers do not make use of malware C&C server to control too many infected machines. Moreover, in our experimental environment consisting of about 10K mobile devices, the distribution of the number of domains queried by internal devices during one day follows heavy-tailed distributions, as shown in Figure 3. There are about half-domains were queried each time. Thus, we can conclude that the number of distinct access devices cannot effectively identify the normal domain. By analyzing APT malware, we find that malicious domain ranked above the top 200,000 [21]. Thus, the number of visitors and the number of pages they visit are a feature used to identify the normal domain. Furthermore, VirusTotal aggregates numerous antivirus products and online scan engine to check for the malicious domain. Thus, we use Alexa ranking and VirusTotal results to judge normal domains, whose Alexa ranking is below 200,000 in international domains and 30,000 in domestic domains, and VirusTotal's test result is less than 3.

4. Building Anomaly Detection

In this section, we explained our anomaly detection algorithm, called GAF.

Definition 1 (global abnormal tree). Let T be the center of a global abnormal tree. N is the number of samples in this global abnormal tree. A test, which consists of d-variate such that the test has a larger distance from T, is an outlier.

Given a dataset $X = (x_1, x_2, \ldots, x_m)$ of m normal samples with d-dimension features, in other words, $x_i = (f_i^1, f_i^2, \ldots, f_i^d)$, the global abnormal tree building process is illustrated as follows. Firstly, we select N normal samples without replacement from the dataset X to build training set $X' = (x_1, x_2, \ldots, x_n)$. Secondly, we calculate the weight of each feature through introducing differentiated information entropy. Thirdly, we select the center of the N normal samples according to

$$T = \left(\sum_{i=1}^{n} \frac{f_i^1}{n}, \sum_{i=1}^{n} \frac{f_i^2}{n}, \ldots, \sum_{i=1}^{n} \frac{f_i^d}{n} \right). \tag{1}$$

An abnormal domain is acquired according to the distance from the node p to the center of the global abnormal tree, which can be calculated using (2). As it is illustrated in (3), once the mean distance of tester is larger than the threshold value T_r, it can be denoted as a suspicious domain.

$$d(p, T) = \sqrt{\sum_{i=1}^{d} \omega_i \left(f_p^i - f_T^i \right)^2 f^i} \tag{2}$$

$$M_d = \frac{\sum_{i=1}^{N} d(p, T_i)}{N} > T_r. \tag{3}$$

In order to identify the weight of each feature, we need to calculate information entropy of each feature using (4), where k represents k distinct values of normal samples in the i_{th} dimension and x_j^i represents the number of normal samples in the i_{th} dimension whose value equals the j_{th} value. Then, each feature splits set into two parts: $\{f^i\}$ and $\{S - f^i\}$. Thus, the information entropy difference is calculated by (5), which

```
Input: $N$: The number of Global Abnormal Tree, $M$: The number of normal sub-samples
       used in each Global Abnormal Tree, $X = (x_1, x_2, \ldots, x_n)$: The normal samples,
       $Y = (y_1, y_2, \ldots, y_k)$: The gery samples
Output: $L$: The list of suspicious domains
(1) For Global Abnormal Tree $T_i$ $(i = 1, 2, \ldots, N)$
(2)     Select $M$ sub-samples from $X$ without replacement: $X_i = (x_1, x_2, \ldots, x_M)$
(3)     Calculate information entropy of each feature $E(f^i)$ $(i = 1, 2, \ldots, d)$
(4)     For each feature $f^i$ $(i = 1, 2, \ldots, d)$
(4.1)       Calculate information entropy difference of each feature $\Delta E(f^i)$ $(i = 1, 2, \ldots, d)$
(4.2)       Set feature weight $\omega_i = \Delta E(f^i)$
(4.3)       Compute standard feature weight $\omega_i$
(5)     Calculate the center of $T_i$ using normalization sub-samples
(6)     Calculate the distance from sample $y_i$ $(i = 1, 2, \ldots, k)$ in $Y$ from the center of $T_i$
(7) End for
(8) Calculate the mean distance $M_d$
(9) Identify abnormal according to $M_d > T_r$
```

Algorithm 1: GAF.

is used to represent feature weight. In (5), the feature weight is normalized.

$$E\left(f^i\right) = \sum_{j=1}^{k} \frac{x_j^i}{n} \log \frac{x_j^i}{n} \tag{4}$$

$$\Delta E\left(f^i\right) = \frac{\sum_{j=1}^{d} E\left(f^j\right)}{n} - \left(E\left(f^i\right) + \frac{\sum_{j=1, j\neq i}^{d} E\left(f^j\right)}{n-1}\right). \tag{5}$$

In the process of anomaly detection based on global outlier factor, the tester is classified as abnormal according to the distance to the center of distinct global abnormal tree. In each tree, the centroid is calculated according to the normal samples selected from training test. And the weight of each feature in the different tree is calculated according to the current normal instances. The pseudocode of GAF algorithm is shown in Algorithm 1.

5. Experiments and Results

In this section, we introduce the experimental setup, the performance metrics, and the obtained results.

5.1. Experimental Setup. In this section, we evaluate the effectiveness of our proposed approach by collecting DNS logs from a network consisting of about 10K mobile devices for 2 weeks. This local area network with high-value information tends to be attacked by APT. Thus, there are many monitor devices deployed at the mobile base station to collect log records, including more than 300,000 DNS requests each day.

Without deploying any filters, it cannot be able to record this large volume of traffic. Hence, the volume of DNS traffic head was restored in log collector to extract DNS logs. The saved field includes source IP, destination IP, domain, query time, and response time. The system had been implemented in Python 3.5, and all experiments were done using an off-the-shelf computer with Intel Core i7 at 3.6 GHz and 16 GB of RAM memory. In order to evaluate the true positive rates and false positive rates of our anomaly detection algorithm, we did the evaluating experiment in our training dataset including part of normal domains from the normal set and malicious domains marked by security experts.

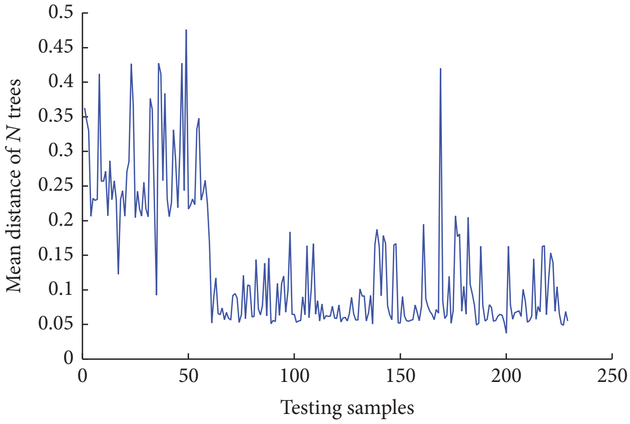


Figure 4: Difference distance between the C&C domains and normal domains.

In our experiment, the parameter $T_r = 0.2$. Almost all of malware domains' mean distance is larger than 0.2, while the mean distance of normal domains is no larger than 0.2 in our testing data. Figure 4 compares the distance between the C&C domains and normal domains. The x-axis represents different testing samples, of which the first 60 are C&C domains, and the back 170 are normal domain names. A noticeable distinction is that almost all of C&C domains' mean distance is larger than 0.2. Meanwhile, Figure 5 illustrates detection performances for malware C&C domain of different threshold. The performances of detection show our

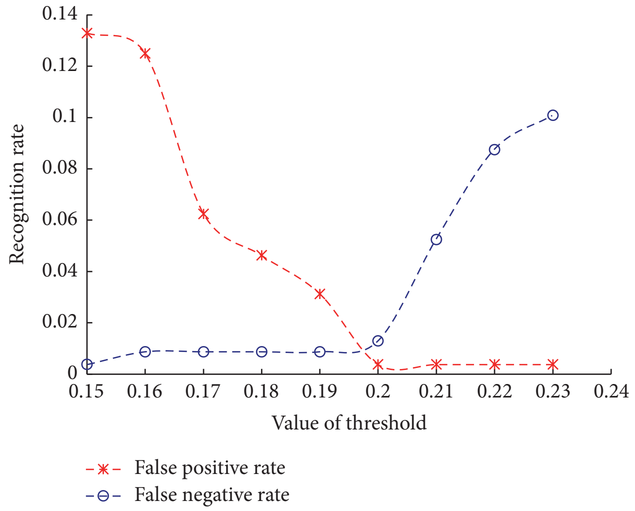


Figure 5: Recognition at different threshold.

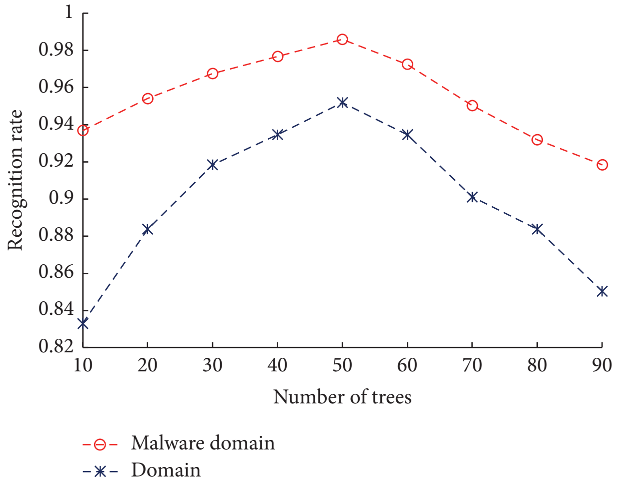


Figure 6: Recognition rate at different number of trees.

anomaly detection algorithm with the lowest false negative rate and false negative rate when the parameter $T_r = 0.2$.

Parameter $N = 50$, $M = 200$. Using the testing data, we have examined the number of trees when N increases from 10 to 90, and the number of samples when M increases from 50 to 450. The results of the experiments are presented by Figures 6 and 7. We made a statistic of recognition rate for a different number of trees and samples. As shown in Figure 6, when N increases from 10 to 50, the percentage of malicious domain identification increases; it is deduced that the scores of the number of trees are greater than 50. This is due to model overfitting. On the other hand, Figure 7 compares the effects of difference number of samples selected by each tree. Overall, when the size of samples is less than 200, false positive rate and false negative rate are decreasing. Thus, the size of samples used in each identification trees is set to 200 and the number of trees is set to 50 in our experimental environment.

The parameters are shown in Table 2.

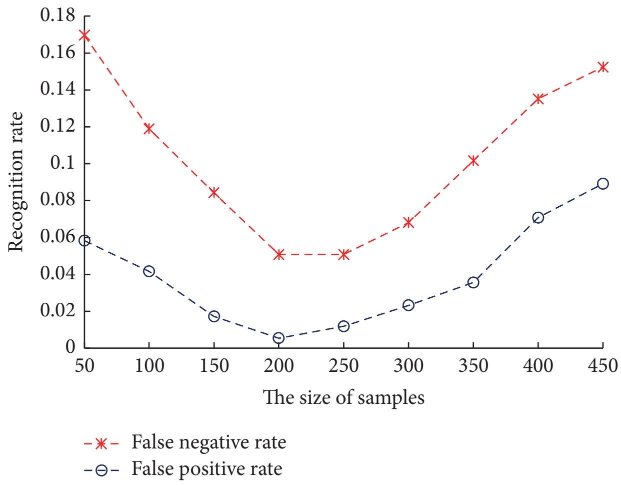


Figure 7: Recognition rate at different size of samples.

Table 2: Experimental parameters settings.

Parameter	Description	Value
T_r	Distance threshold	0.2
N	Number of trees	50
M	Number of samples	200

5.2. Results of Experiments and Discussion. The detection performances of APT malware C&C domain are expressed by performance metrics that describes both accuracy and time requirements of different detection algorithms. The accuracy is expressed by following metrics:

(1) False Recognition Rate: $\mathrm{FR} = \mathrm{FN}_n/(\mathrm{TP}_n + \mathrm{FN}_n)$

(2) Precision: $\mathrm{Pr} = \mathrm{TP}_n/(\mathrm{TP}_n + \mathrm{FP}_n)$

(3) Recall Rate: $R = \mathrm{TP}_n/(\mathrm{TP}_n + \mathrm{FN}_n)$

(4) F-Measure: $F\text{-}M = 2 \times \mathrm{Pr} \times R/(\mathrm{Pr} + R)$

In the above equations, TP_n refers to the number of normal domain names that are recognized as normal domain names, TN_n refers to the number of malicious domain names that are recognized as malicious domain names, FP_n refers to the number of malicious domain names that have been mistaken for normal domain names, and FN_n refers to the number of normal domain names that are incorrectly identified as normal domain names, respectively. Thus, the higher the value of Pr, R, and $F\text{-}M$, the better the recognition effect of anomaly detection algorithms. Conversely, the lower the value of FR, the better the performance.

Some experiments were performed to evaluate the performance of our proposed approach for detection APT malware C&C domains. Table 3 presents the results of different anomaly detection algorithms. GAF with information entropy yielded average detection accuracy of 98.3 percent and standard GAF yielded an average detection accuracy of 93.9 percent. Also, GAF with information entropy yielded an FP rate and FN rate of 0.013 and 0.004 percent, respectively, while standard GAF yielded an FP rate and FN rate of 0.056 and 0.004 percent, respectively. Additionally, GAF with

Table 3: Detection accuracy of different algorithms.

Algorithms	Items			
	APA	FP	FN	T (second)
iForest	0.883	0.052	0.065	17
LOF	0.765	0.169	0.109	973
KNN	0.674	0.2	0.126	4573
GAF (with information entropy)	0.983	0.013	0.004	18
GAF	0.939	0.056	0.004	15

Notes. APA, overall recognition rate; FP, false positive rate; FN, false negative rate; T, time.

Table 4: Empirical comparison of different number of trees.

Algorithms	Items			
	FR	Pr	R	F-M
iForest	0.088	0.928	0.912	0.92
LOF	0.147	0.788	0.853	0.853
KNN	0.17	0.754	0.83	0.83
GAF (with information entropy)	0.0058	0.98	0.994	0.994
GAF	0.0058	0.928	0.928	0.994

information entropy and standard GAF yielded a detection speed of 18.7 seconds and 15.6 seconds, respectively. These results revealed that the overall performance of GAF with information entropy outperformed standard GAF, implying that feature weight is a better optimization parameter.

Additionally, as shown in Table 3, GAF with information entropy was compared to three traditional anomaly detection algorithms and a detection accuracy of 98.3 percent was achieved, which is higher than the three detection accuracies (i.e., 88.3, 76.5, and 67.4 percent). Also, GAF with information entropy performed better in terms of time compared to LOF and KNN with more than 16 minutes.

Results from the experiments were compared to results of different anomaly detection algorithms. As shown in Table 4, GAF (with information entropy) has the highest PR, R, and F-M and the lowest FR. The R value of our proposed GAF algorithm reaches 0.994, which is higher than other algorithms. The F-M value and R value of GAF are higher than other three traditional algorithms. The F-M value and FR value of GAF and GAF with information entropy are the same. That was because the feature has no effect on normal sample identification. However, the PR value of GAF algorithm using differentiated information entropy to represent the weight of different features is higher than GAF whose feature has the same effect in identifying domains. Since some normal domains overlap with malware C&C domains in the feature space, LOF and KNN using all the normal samples have higher false negative rate and false positive rate. Moreover, iForest using depth of trees has certain assumptions. In our work, there are three malicious domains not yet identified since their behaviors are the same as the normal domain. The root cause of the false positives is anomaly detection.

6. Related Work

The proposed approach combines statistical knowledge related to malware using DNS to locate C&C servers with anomaly detection. Thus, the main motivation behind our work relies on APT detection, anomaly detection, DNS malicious domain detection, and botnet detection.

APT Detection. Siddiqui et al. [22] proposed a fractal based APT anomalous patterns classification method with the goal of reducing both false positives and false negatives using various features of a TCP/IP connection. Marchetti et al. [23] identified and ranked suspicious hosts possibly involved in data exfiltrations related to APT according to suspiciousness score for each internal host. Mcafee [24] extracted network features of several APT malware to identify APT C&C communication traffic. IDns [25] analyzed a large volume of DNS traffic and network traffic of suspicious malware C&C server to detect APT malware infection. Unfortunately, these approaches identified APT after data exfiltrations. Our proposed approach identifies APT malware in the stage of establishing C&C channel.

Wang et al. [26] made use of independent access to find out HTPP-based C&C domain. Barceló-Rico et al. [27] developed a semisupervised classification system to detect suspicious instances for identifying APT attacks based on HTTP traffic. However, they cannot effectively identify malware C&C domain based on other protocols. Our proposed approach uses mobile DNS logs to identify APT malware that utilizes DNS to support their C&C infrastructure.

Friedberg et al. [28] proposed an anomaly detection system to identify APT according to security logs from individual hosts. But host logs were often impractical to obtain. Bertino and Ghinita [29] detected APT related to data exfiltrations by analyzing DataBase Management System (DBMS) access logs. Liu et al. [30] made use of network traffic to identify data exfiltrations based on automatic signature generation but cannot apply even if the attacker uses encrypted communications and standard protocols. Our proposed approach identifies APT malware prior to data exfiltrations and use partial data to reduce storage overhead.

DNS Malicious Domain Detection. In order to judge whether a new domain is malicious or not, Notos [31] constructed the network, zone, and evidence-based features to compute reputation scores for new domains. However, it was dependent on large amounts of historical maliciousness data. Exposure [32] employed large-scale, passive DNS analysis techniques to detect domains that are involved in malicious activity. Unfortunately, it relied on prior knowledge of label malware C&C domain in the training phase. Notes [31] and Exposure [32] identify malicious domains based on DNS traffic from local recursive DNS servers. Unfortunately, it identified malicious domains that are misused in a variety of malicious activity. Our proposed detection approach focuses on APT malware. Other related work used graph-based inference technique to discover new malicious domains. Manadhata et al. [33] constructed a host-domain graph to detect malicious domains combined with belief propagation. Rahbarinia et al. [34] built a machine-to-domain bipartite graph to efficiently detect new malware-control domain by tracking the DNS query behavior. Khalil et al. [35] developed graphs reflecting the global correlations among domains to discover malicious domain based on their topological connection to known malicious domains. However, those methods required prior knowledge that known partial domain names.

Botnet Detection. Botnet detection is also interesting related work to compare the problem of APT malware C&C domain detection. Sniffer [3] and BotMiner [5] detected botnet hosts based on the similarity of connections. BotGAD [4] also detected botnet from the group activity characteristics in network traffic. However, the above-mentioned detection approaches are difficult for detecting APT with limited communication samples and small-scale victims.

7. Conclusion

APT malware identification is still a challenge to network security since few attacks traces exist in mass behaviors. Most malware makes use of domain name to locate C&C server. Thus, C&C domain detection by analyzing DNS records is feasible. This paper proposes an efficient APT malware C&C domain detection approach capable of handling unmarked data. In our proposed anomaly detection algorithm, information entropy is introduced to indicate the different influence of each feature. The anomaly detector was evaluated on a dataset consisting of more than 300,000 DNS requests each day during two weeks from a mobile station. The experimental results show that our proposed approach can produce an overall R and FM coefficient of 0.994. This reveals that GAF has the highest detection accuracy rate. Moreover, our approach is applicable to the real environment without domain category.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61572115) and the Key Basic Research of Sichuan Province (Grant no. 2016JY0007).

References

[1] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 International Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings*, vol. 8735 of *Lecture Notes in Computer Science*, pp. 63–72, Springer, Berlin, Germany, 2014.

[2] M. Ask, P. Bondarenko, J. E. Rekdal et al., "Advanced Persistent Threat (APT) beyond the hype," in *Project Report in IMT4582 Network Security at GjoviN University College*, Springer, Berlin, Germany, 2013.

[3] 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*, 2008.

[4] H. Choi, H. Lee, and H. Kim, "BotGAD: detecting botnets by capturing group activities in network traffic," in *Proceedings of the 4th International ICST Conference on Communication System Software and Middleware (COMSWARE '09)*, June 2009.

[5] G. Gu, R. Perdisci, J. Zhang et al., "BotMiner: clustering analysis of network traffic for protocol-and structure-independent botnet detection," *USENIX Security Symposium*, vol. 5, no. 2, pp. 139–154, 2008.

[6] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware C8C detection," *ACM Computing Surveys*, vol. 49, no. 3, article 59, 2016.

[7] K. Zetter, "Google hack attack was ultra sophisticated, new details show," *Wired Magazine*, vol. 14, 2010.

[8] Y. Zhang, C. Xu, H. Li, and X. Liang, "Cryptographic public verification of data integrity for cloud storage systems," *IEEE Cloud Computing*, vol. 3, no. 5, pp. 44–52, 2016.

[9] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang, "SCLPV: secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 159–170, 2015.

[10] R. Sonawane, T. Tajane, P. Chavan et al., "Anomaly based intrusion detection network system," *Software Engineering and Technology*, vol. 8, no. 3, pp. 66–69, 2016.

[11] M. Wan, L. Li, J. Xiao, C. Wang, and Y. Yang, "Data clustering using bacterial foraging optimization," *Journal of Intelligent Information Systems*, vol. 38, no. 2, pp. 321–341, 2012.

[12] X. Liu, X. Zhang, Y. Jiang, and Q. Zhu, "Modified t-distribution evolutionary algorithm for dynamic deployment of wireless sensor networks," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 6, pp. 1595–1602, 2016.

[13] N. Suryavanshi and A. Jain, "Phishing detection in selected feature using modified SVM-PSO," *International Journal of Research in Computer and Communication Technology*, vol. 5, no. 4, pp. 208–214, 2016.

[14] W. Wang, L. Li, H. Peng, J. Xiao, and Y. Yang, "Synchronization control of memristor-based recurrent neural networks with perturbations," *Neural Networks*, vol. 53, pp. 8–14, 2014.

[15] M. X. Ma, H. Y. Ngan, and W. Liu, "Density-based outlier detection by local outlier factor on largescale traffic data," *Electronic Imaging*, vol. 2016, no. 14, pp. 1–4, 2016.

[16] J. A. Khan and N. Jain, "Improving intrusion detection system based on KNN and KNN-DS with detection of U2R, R2L attack for network probe attack detection," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 5, pp. 209–212, 2016.

[17] L. Sun, S. Versteeg, S. Boztas, and A. Rao, "Detecting anomalous user behavior using an extended isolation forest algorithm: an enterprise case study," https://arxiv.org/abs/1609.06676.

[18] P. Singh Chahal and S. Singh Khurana, "TempR: application of stricture dependent intelligent classifier for fast flux domain detection," *International Journal of Computer Network and Information Security*, vol. 8, no. 10, pp. 37–44, 2016.

[19] A. R. Kang, J. Spaulding, and A. Mohaisen, "Domain name system security and privacy: old problems and new challenges," https://arxiv.org/abs/1606.07080.

[20] B. Yu, L. Smith, and M. Threefoot, "Semi-supervised time series modeling for real-time flux domain detection on passive DNS traffic," in *Machine Learning and Data Mining in Pattern Recognition: 10th International Conference, MLDM 2014, St. Petersburg, Russia, July 21–24, 2014. Proceedings*, vol. 8556 of *Lecture Notes in Computer Science*, pp. 258–271, Springer International Publishing, 2014.

[21] Alexa Web Information Company, 2015, http://www.alexa.com/topsites.

[22] S. Siddiqui, M. S. Khan, K. Ferens, and W. Kinsner, "Detecting advanced persistent threats using fractal dimension based machine learning classification," in *Proceedings of the 2nd ACM International Workshop on Security and Privacy Analytics (IWSPA '16)*, pp. 64–69, ACM, 2016.

[23] M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for Advanced Persistent Threat detection," *Computer Networks*, vol. 109, pp. 127–141, 2016.

[24] N. Villeneuve and J. Bennett, *Detecting Apt Activity with Network Traffic Analysis*, Trend Micro Incorporated, 2012, http://www.trendmicro.pl/cloud-content/us/pdfs/security-intelligence/white-papers/wp-detecting-apt-activity-with-network-traffic-analysis.pdf.

[25] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT malware infections based on malicious DNS and traffic analysis," *IEEE Access*, vol. 3, pp. 1132–1142, 2015.

[26] X. Wang, K. Zheng, X. Niu, B. Wu, and C. Wu, "Detection of command and control in advanced persistent threat based on independent access," in *Proceedings of the IEEE International Conference on Communications (ICC '16)*, pp. 1–6, Kuala Lumpur, Malaysia, May 2016.

[27] F. Barceló-Rico, A. I. Esparcia-Alcázar, and A. Villalón-Huerta, "Semi-supervised classification system for the detection of advanced persistent threats," in *Recent Advances in Computational Intelligence in Defense and Security*, pp. 225–248, Springer International Publishing, 2016.

[28] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: from network event correlation to incident detection," *Computers and Security*, vol. 48, pp. 35–57, 2015.

[29] E. Bertino and G. Ghinita, "Towards mechanisms for detection and prevention of data exfiltration by insiders," in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security (ASIACCS '11)*, pp. 10–19, ACM, March 2011.

[30] Y. Liu, C. Corbett, K. Chiang, R. Archibald, B. Mukherjee, and D. Ghosal, "SIDD: a framework for detecting sensitive data exfiltration by an insider attack," in *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences (HICSS '09)*, pp. 1–10, January 2009.

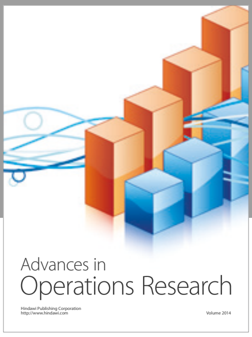
[31] M. Antonakakis, R. Perdisci, D. Dagon et al., *Notos: Building a Dynamic Reputation System for DNS*, Georgia Institute of Technology College of Computing, Atlanta, Ga, USA, 2010.

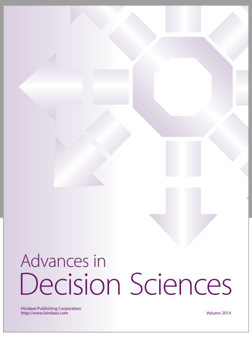
[32] L. Bilge, E. Kirda, C. Kruegel et al., "EXPOSURE: finding malicious domains using passive DNS analysis," in *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS '11)*, 2011.

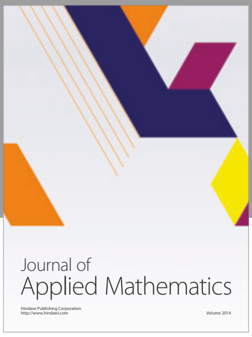
[33] P. K. Manadhata, S. Yadav, P. Rao, and W. Horne, "Detecting malicious domains via graph inference," in *Computer Security—ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7–11, 2014. Proceedings, Part I*, vol. 8712 of *Lecture Notes in Computer Science*, pp. 1–18, Springer International Publishing, 2014.

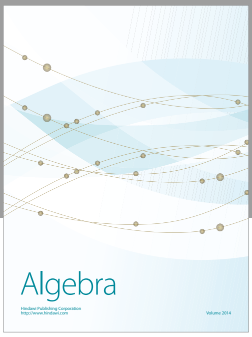
[34] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: efficient behavior-based tracking of malware-control domains in large ISP networks," in *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '15)*, pp. 403–414, June 2015.

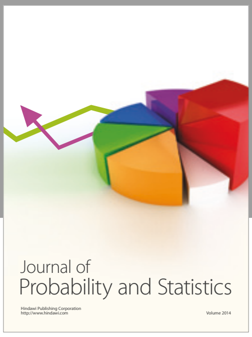
[35] I. Khalil, T. Yu, and B. Guan, "Discovering malicious domains through passive DNS data graph analysis," in *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security (ASIA CCS '16)*, pp. 663–674, ACM, June 2016.


Advances in
Operations Research

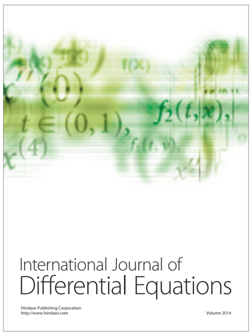

Advances in
Decision Sciences

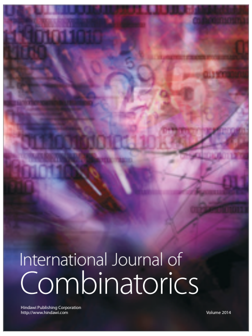

Journal of
Applied Mathematics

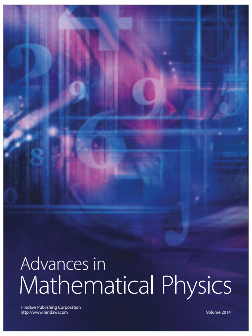

Algebra

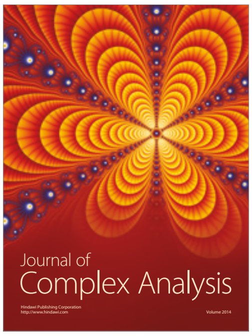

Journal of
Probability and Statistics

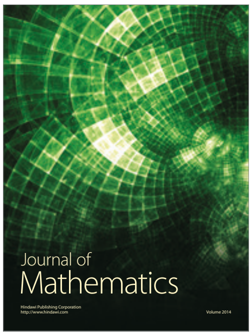
The Scientific
World Journal

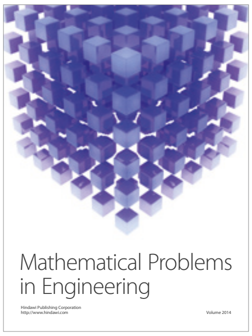

International Journal of
Differential Equations

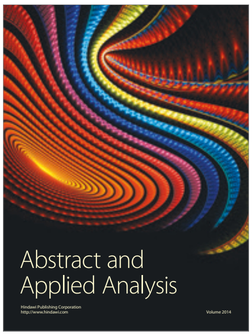

International Journal of
Combinatorics

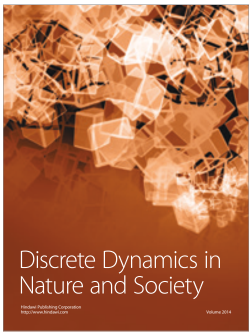

Advances in
Mathematical Physics

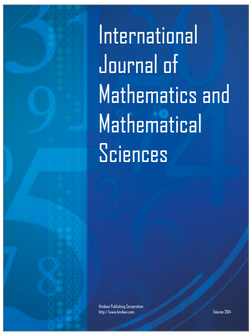

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

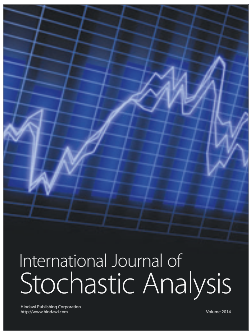

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization