九州工業大学学術機関リポジトリ
Kyutacar
Kyushu Institute of Technology Academic Repository

Clustering Malicious DNS Queries for Blacklist-Based Detection

著者	Satoh Akihiro, Nakamura Yutaka, Nobayashi Daiki, Sasai Kazuto, Kitagata Gen, Ikenaga Takeshi
journal or publication title	IEICE Transactions on Information and Systems
volume	E102.D
number	7
page range	1404-1407
year	2019-07-01
URL	http://hdl.handle.net/10228/00007647

doi: info:doi/10.1587/transinf.2018EDL8211

IEICE TRANS. INF. & SYST., VOL.E102–D, NO.7 JULY 2019

LETTER

Clustering Malicious DNS Queries for Blacklist-Based Detection*

Akihiro SATOH[†a)], **Yutaka NAKAMURA**[†], **Daiki NOBAYASHI**[†], **Kazuto SASAI**[††], **Gen KITAGATA**[†††], ***and*** **Takeshi IKENAGA**[†], ***Members***

SUMMARY Some of the most serious threats to network security involve malware. One common way to detect malware-infected machines in a network is by monitoring communications based on blacklists. However, such detection is problematic because (1) no blacklist is completely reliable, and (2) blacklists do not provide the sufficient evidence to allow administrators to determine the validity and accuracy of the detection results. In this paper, we propose a malicious DNS query clustering approach for blacklist-based detection. Unlike conventional classification, our cause-based classification can efficiently analyze malware communications, allowing infected machines in the network to be addressed swiftly.

key words: *malware, blacklist, DNS query, machine learning*

1. Introduction

Some of the most serious threats to network security involve malware. Cyber-criminals use malware-infected machines to undertake malicious activities such as stealing confidential information, distributing malware, and sending spam. According to a McAfee report [1], over 300,000 new forms of malware are created each day. To prevent damage from malware, administrators must swiftly identify and remove infected machines that reside in their networks. One common way to detect infected machines in a network is by monitoring communications based on blacklists. To improve the detection capability, several studies have attempted to automatically update blacklist entries by using machine learning techniques [2]. However, such detection is problematic because (1) no blacklist is completely reliable, and (2) blacklists do not provide the sufficient evidence to allow administrators to determine the validity and accuracy of the detection results. Therefore, simply matching communications with blacklist entries is insufficient, and administrators should pursue their causes by investigating the communications themselves.

The goal of this study is to classify malicious DNS queries detected through blacklists by their causes. We focus on DNS because domain name resolution always occurs prior to malware communications. This cause-based classification drastically reduces the number of malicious DNS queries to be investigated because the investigation scope is limited to only representative queries in classification results. The remainder of this paper is organized as follows: In Sect. 2, we review the related studies and their limitations. In Sect. 3, we propose a malicious DNS query clustering approach for blacklist-based detection. We describe experiments conducted to analyze the effectiveness of our approach in classifying malicious DNS queries in Sect. 4. Finally, we summarize our conclusions and future work in Sect. 5.

Manuscript received October 10, 2018.
Manuscript revised December 28, 2018.
Manuscript publicized April 5, 2019.

[†]The authors are with Kyushu Institute of Technology, Kitakyushu-shi, 804–8550 Japan.
[††]The author is with Ibaraki University, Hitachi-shi, 316–8511 Japan.
[†††]The author is with Tohoku University, Sendai-shi, 980–8577 Japan.

*This work was supported by JSPS KAKENHI Grant Number JP18K11296. Part of this work was carried out under the Cooperative Research Project of the RIEC, Tohoku University.

a) E-mail: satoh@isc.kyutech.ac.jp

DOI: 10.1587/transinf.2018EDL8211

2. Related Work

Kheir et al. [3] showed that blacklists typically contain a considerable number of errors. These errors stem from domains containing mixed benign and malicious codes, such as cloud services, advertising networks, and dynamic DNSs. Automatically generated blacklists simply exacerbate this problem. The authors of [3] attempted to improve the detection accuracy by cross-checking domains on multiple blacklists. However, a blacklist has specific coverage; consequently, cross-checking greatly narrows the coverage.

Kührer et al. [4] evaluated the effectiveness of 19 types of blacklists by considering factors, such as unregistered domains, parking domains, and sinkhole domains. In the evaluation, the authors of [4] used only datasets prelabeled as benign or malicious. This is because the investigation scope expands proportionally to the number of detections, which complicates determining the validity and accuracy of detection results.

Some previous studies have focused on analyzing DNS queries and responses. Wang et al. [5] developed a system called DBod that detects and classifies infected machines based on the statistical similarity of query behaviors. However, DBod is specific to domain generation algorithm-based malware and cannot be applied to other kinds of malware. Berger et al. [6] developed a system called DNSMap that discovers potentially compromised machines by rapidly changing DNS queries. DNSMap derives the similarity of character strings in domain names by considering their hierarchical structure but the superficial similarity does not perform sufficiently well in classifying malicious queries by

Copyright © 2019 The Institute of Electronics, Information and Communication Engineers

their causes.

3. Proposal

In this paper, we propose a malicious DNS query clustering approach for blacklist-based detection. This approach focuses on the following observation: a malware communication is divided into several transactions; thus surrounding queries that occur before and after a malicious query detected through blacklists help in estimating the cause of the malicious query. By numerically comparing their surrounding queries, the approach is able to classify malicious queries by their causes. Unlike conventional classification, which is based on the superficial similarity of character strings in domain names, our cause-based classification can efficiently analyze malware communications, allowing infected machines in the network to be addressed swiftly.

General classification techniques in natural language processing can be probably applied by regarding a malicious query and surrounding queries as words. Le et al. [7] proposed Doc2Vec that classifies various documents by using co-occurrences between words. Unfortunately, the performance deteriorates due to the influence of irrelevant queries to classification. In contrast, our approach weights queries based on insight about malware communications.

Figure 1 shows an overview of the proposed approach, which has three functions: (1) Query Sub-log Selection, (2) Numerical Representation, and (3) Similarity Calculation. The following sections describe each function in detail.

3.1 Query Sub-Log Selection

A query log for the input of our approach is a record of queries for domain name resolution to a recursive DNS from machines on a network. In the query log, each query has values such as a timestamp, source address, and queried domain name along with class and type. In particular, note that domain names in the query log are shortened to and replaced to primary domain names. A primary domain is the highest-level domain name given to a registrar. For example, the primary domain names for `www.ieice.org` and `smtp.kyutech.ac.jp` would respectively be `ieice.org` and `kyutech.ac.jp`.

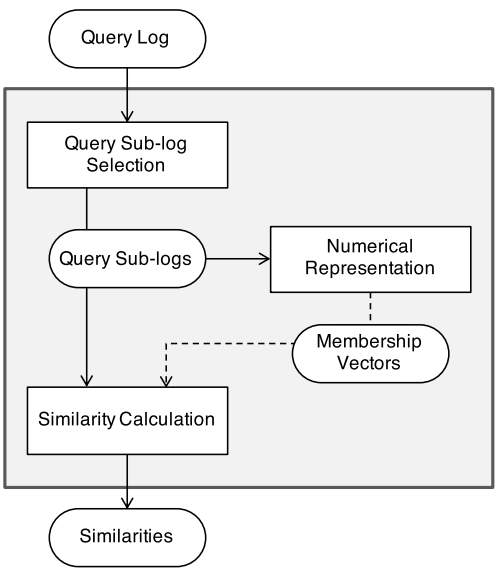


Fig. 1 Overview of the malicious DNS query clustering approach for blacklist-based detection.

First, this function detects malicious queries from a query log through comparison with blacklists. A query x_n in the query log is considered to be malicious when the domain name in query x_n matches the entry in blacklist L_B. The function then selects all the queries with the same source address as malicious query x_n that occur within t_α seconds before and after malicious query x_n. These queries form the query sub-log X_n. This step is conducted because these queries in query sub-log X_n help in estimating the cause of malicious query x_n. Finally, the output of the function is set $\mathbb{X}$, which consists of the N query sub-logs, where N denotes the number of malicious queries detected in the query log.

3.2 Numerical Representation

This function attempts to numerically represent queries based on their co-occurrences by using the two machine learning techniques: Word2Vec [8] and soft clustering with Gaussian Mixture Models [9]. Through this step, effective features are extracted from the enormous number of queries included in all the query sub-logs.

First, the function applies a Word2Vec model to all the query sub-logs $\mathbb{X}$ to create a distributed representation based on co-occurrences between queries. A distributed representation is to associate one data with one point in multi-dimensional space, and Word2Vec, which has drawn considerable attention in the field of natural language processing, expresses the features of each word as a vector based on the assumption that each word in a sentence has a strong relation with its surrounding words. We modify Word2Vec to change its focus from words in a sentence to queries in a query sub-log as follows: (1) we replace words with queried domain names, and (2) whereas the conventional Word2Vec algorithm uses the distance between words in a sentence to measure co-occurrences, we instead use time interval between queries in a query sub-log, where the time interval between queries is restricted to within t_β seconds.

Next, the function applies soft clustering with Gaussian Mixture Models to the distributed representation of queries. Soft clustering yields the probability that each data point belongs to each cluster. Because each cluster consists of queries with similar co-occurrences in the clustering results, a cluster can be expected to indicate a transaction in a malware communication. Finally, the output of the function is the membership vector of each query, written as follows:

$$\vec{p}(x_i) = \left(p(c_1|x_i), \cdots, p(c_m|x_i), \cdots, p(c_M|x_i) \right),$$

where M is the number of clusters and $p(c_m|x_i)$ is the probability that query x_i belongs to cluster c_m.

3.3 Similarity Calculation

This function calculates the feature vector from the membership vectors of the queries found in each query sub-log. By comparing the similarity of feature vectors according to their cosine distance, the function achieves cause-based classification latently indicated by the malicious queries and their surrounding queries. Note that we emphasize queries commonly appearing in multiple query sub-logs based on the following insight about malware communications: infected machines in a same malware family repeatedly communicate with a same malicious domain group.

The membership vectors of the queries found in each query sub-log imply the transactions constituting a malware communication; in query sub-logs, the similarities of transactions are strongly dependent on the similarities of causes; thus, the feature vector for each query sub-log is derived from the weighted sum of the membership vectors of the queries found in each query sub-log, as follows:

$$\vec{X}_n = \sum\nolimits_{x_i \in X_n} w_\alpha(x_i)\ w_\beta(x_i)\ w_\gamma(x_i)\ \vec{p}(x_i)\ .$$

Here, $w_\gamma = 0$ if the domain name for query x_i is included in whitelist L_W; otherwise, $w_\gamma = 1$. From the insight concerning malware communications, the weights w_α and w_β for query x_i are respectively defined as

$$w_\alpha(x_i) = \frac{|\mathcal{F}_{addr}(x_i, \mathbb{X}) \cap \mathcal{F}_{name}(x_i, \mathbb{X})|}{|\mathcal{F}_{addr}(x_i, \mathbb{X})|}$$

and

$$w_\beta(x_i) = \frac{|\mathcal{F}_{list}(x_i, \mathbb{X}) \cap \mathcal{F}_{name}(x_i, \mathbb{X})|}{|\mathcal{F}_{list}(x_i, \mathbb{X})|}\ ,$$

where $\mathcal{F}_{name}(x_i, \mathbb{X})$, $\mathcal{F}_{addr}(x_i, \mathbb{X})$, and $\mathcal{F}_{list}(x_i, \mathbb{X})$ are differing subsets of set $\mathbb{X}$. $\mathcal{F}_{name}(x_i, \mathbb{X})$ is a set formed from query sub-logs including a query with the same domain name as query x_i; $\mathcal{F}_{addr}(x_i, \mathbb{X})$ is a set formed from query sub-logs that consist of queries with the same source address as query x_i; $\mathcal{F}_{list}(x_i, \mathbb{X})$ is a set formed from query sub-logs that are detected by the same blacklist entry as query sub-log X_n including query x_i. Additionally, $|\cdot|$ indicates the number of set elements. Finally, the output of the function is the similarities of malicious queries calculated by comparing the feature vectors of query sub-logs through their cosine distance.

4. Evaluation

In this section, we evaluate the effectiveness of the proposed approach through experiments using DNS queries observed on a campus network. This evaluation focuses primarily on the classification accuracy of malicious queries and the efficiency of the analysis of their causes. We first describe the experimental setup in Sect. 4.1 and then discuss the experimental results in Sect. 4.2.

4.1 Experimental Setup

The dataset used for the experiments consisted of DNS queries observed from January 2017 to February 2018 on a campus network. For blacklist L_B, we employed three public listings of malicious domains [10]–[12]; for whitelist L_W, we employed the top one million domains provided by Alexa [13].

We set the parameters in our approach to the following values: $t_\alpha = 90$ and $t_\beta = 1.0$. The number of iterations, number of dimensions, and learning rate for Word2Vec were set to 250,000, 100, and 0.0005, respectively. By experimentally verifying several values, we determined the above five parameters. We will address parameter optimization in future work. In soft clustering, Bayesian Information Criterion (BIC) was used for model decision. The parameters were selected according to BICs of the 14 types of multivariate mixture models and number of clusters up to 20. For further details refer to [9].

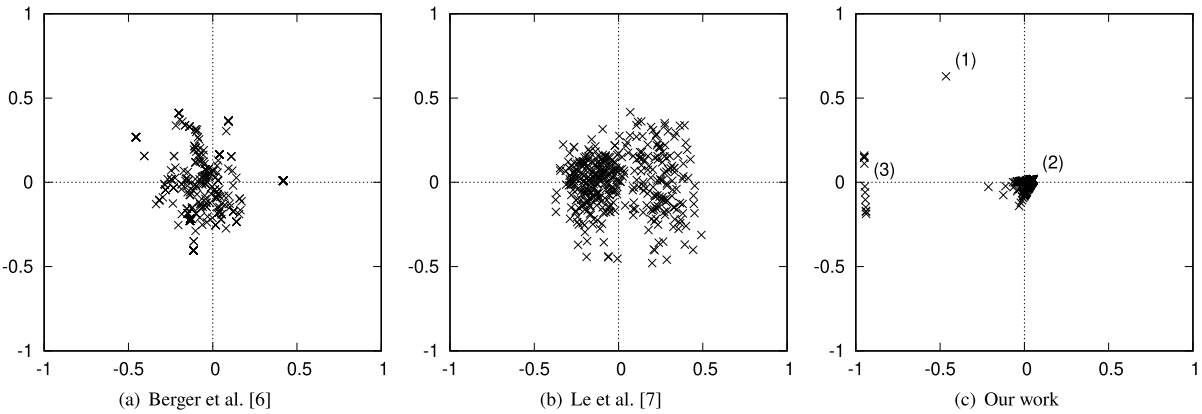
For comparison with the proposed approach, we implemented the two different approaches for classifying malicious queries described in [6] and [7]. The first implementation uses the similarity of character strings in domain names, and the second implementation uses Doc2Vec, which is a well-known extension to Word2Vec. While Word2Vec derives the feature vectors of words, Doc2Vec derives the feature vectors of documents. We set the maximum distance between words for measuring co-occurrences to 5 for Doc2Vec; the other parameters were set to the same values as those used for the proposed approach.

4.2 Discussion

The authors of [14] reported that many types of malware communicate through TXT-type queries. Therefore, we considered TXT-type queries in which the domain names matched blacklist entries to be malicious. By matching these criteria, 388 queries with 158 unique domains were detected from the dataset.

Figure 2 shows the experimental results, using multidimensional scaling to visualize the similarity among malicious queries. In the figure, each symbol represents a malicious query, and the distance between symbols indicates the similarity between malicious queries. Because the symbols in Figs. 2 (a) and 2 (b) are scattered, it is difficult to determine the similarity of the malicious queries. The respective reasons for performance deterioration in the two compared implementations are (1) the limitations of classifying malicious queries based on the superficial similarity of character strings in domain names, and (2) the influence of surrounding queries before and after malicious queries that are unnecessary for estimating their causes. In contrast, our approach clearly classifies the 388 malicious queries detected through blacklists into 3 clusters, suggesting the possibility for efficient analysis.

The numbers of malicious queries classified into clus-



(a) Berger et al. [6] (b) Le et al. [7] (c) Our work

Fig. 2 Experimental results.

ters (1), (2), and (3) in Fig. 2 (c) were 1, 375, and 12, respectively. Only one malicious query was classified into cluster (1) because nearly no surrounding queries occurred either before or after the malicious query, i.e., only seven queries to two domains were observed during the period. In this case, it is difficult for the proposed approach to correctly derive similarity. In cluster (2), queries related to domain reputation frequently occurred before and after the malicious queries, for example, to `spamhaus.org`, `abuseat.org`, and `barracudacentral.org`. Accordingly, we believe that the malicious queries in cluster (2) were caused by misdetection of communications from some security appliances. In cluster (3), queries to BitTorrent tracking sites occurred before and after the malicious queries, for example, to `opentrackr.org`, `asnet.pw`, and `blackunicorn.xyz`. The communications were to domains included in the blacklists, and several studies reported malware that use P2P for interactions [15], [16]. Accordingly, we attribute the malicious queries in cluster (3) to be due to malware infection. The results confirmed that each cluster consists of malicious queries with a common cause, which suggests the possibility for accurate classification.

Concluding the evaluation, our approach realizes to classify the 388 malicious queries detected through blacklists into 3 clusters, and each cluster consists of malicious queries with a common cause. These results indicate that administrators can pursue all the causes by investigating only representative queries of each cluster, and thereby swiftly address infected machines in the network.

5. Conclusions

In this study, we aimed to classify malicious DNS queries detected through blacklists by their causes. Through the experiments, we confirmed that our approach could group the 388 malicious queries into the 3 clusters consisting of queries with a common cause. By enabling administrators to swiftly address infected machines in the network, our approach contributes to dramatically improving network security.

In future work, we plan to evaluate the proposed approach using non-TXT-type queries detected through blacklists. We will also consider adding a new function to remove clusters unrelated to malware communications.

References

[1] J.A. Lewis, "Economic impact of cybercrime — At $600 billion and counting – No slowing down," https://www.csis.org/analysis/economic-impact-cybercrime, 2018.
[2] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Efficient and accurate behavior-based tracking of malware-control domains in large ISP networks," ACM Trans. Privacy and Security, vol.19, no.2, pp.4:1–4:31, 2016.
[3] N. Kheir, F. Tran, P. Caron, and N. Deschamps, "Mentor: Positive DNS reputation to skim-off benign domains in botnet C&C blacklists," Proc. International Conference on ICT Systems Security and Privacy Protection, pp.1–14, 2014.
[4] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," Proc. International Symposium on Research in Attacks, Intrusions and Defenses, Lecture Notes in Computer Science, vol.8688, pp.1–21, Springer, Cham, 2014.
[5] T.-S. Wang, H.-T. Lin, W.-T. Cheng, and C.-Y. Chen, "DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis," Computers & Security, vol.64, pp.1–15, 2017.
[6] A. Berger, A. D'Alconzo, W.N. Gansterer, and A. Pescapé, "Mining agile DNS traffic using graph analysis for cybercrime detection," Computer Networks, vol.100, pp.28–44, 2016.
[7] Q. Le et al., "Distributed representations of sentences and documents," Proc. International Conference on Machine Learning, pp.1188–1196, 2014.
[8] T. Mikolov et al., "Distributed representations of words and phrases and their compositionality," Advances in Neural Information Processing Systems, pp.3111–3119, 2013.
[9] L. Scrucca, M. Fop, T.B. Murphy, and A.E. Raftery, "mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models," The R Journal, vol.8, no.1, pp.289–317, 2016.
[10] DNS-BH, https://www.malwaredomains.com
[11] hpHosts Online, https://hosts-file.net
[12] abuse.ch, https://abuse.ch
[13] Alexa, http://www.alexa.com
[14] H. Ichise, Y. Jin, and K. Iida, "Analysis of DNS TXT record usage and consideration of botnet communication detection," IEICE Trans. Commun,, vol.E101-B, no.1, pp.70–79, Jan. 2018.
[15] R. Cuevas, M. Kryczka, R. González, A. Cuevas, and A. Azcorra, "TorrentGuard: Stopping scam and malware distribution in the BitTorrent ecosystem," Computer Networks, vol.59, pp.77–90, 2014.
[16] A.D. Berns et al., "Searching for malware in BitTorrent," University of Iowa Computer Science Technical Report, pp.1–10, 2008.