Malicious Domain Detection Using Statistical Features of Domain Strings, Public Information, and DNS Logs

Reo Kusumi
Graduate School of Science and Engineering
Kindai University
Higashi-Osaka, 577-8502 Japan
2433340472w@kindai.ac.jp

Makoto Takita
Graduate School of Engineering
Kobe University
Kobe, 657-8501 Japan
https://orcid.org/0000-0002-2532-6416

Thin Tharaphe Thein
Graduate School of Engineering
Kobe University
Kobe, 657-8501 Japan
https://orcid.org/0000-0002-1213-0393

Masami Mohri
Faculty of Informatics & Cyber Informatics Research Institute
Kindai University
Higashi-Osaka, 577-8502 Japan
https://orcid.org/0000-0002-4644-4550

Yoshiaki Shiraishi
Graduate School of Engineering
Kobe University
Kobe, 657-8501 Japan
https://orcid.org/0000-0002-8970-9408

***Abstract*—Domain names, which are indispensable for accessing Internet services, are also exploited by attackers to direct users to phishing sites or malware-infected sites. Therefore, determining whether a domain name is malicious or not is crucial for security measures. In this paper, we perform malicious domain detection using a publicly available dataset that includes statistical features of domain strings, publicly available information from sources such as WHOIS, and DNS query and response information. As a result, we achieve a recall of 0.940 and an F1-score of 0.968 with a Random Forest classifier, a recall of 0.946 and an F1-score of 0.968 with both Random Forest and XGBoost classifiers using only statistical features, and a recall of 0.983 and an F1-score of 0.986 with an XGBoost classifier using statistical features and DNS query and response information.**

***Index Terms*—machine learning, malicious domain, dataset, domain name, domain generation algorithm**

I. Introduction

A Domain Generation Algorithm (DGA) generates a large number of domains, enabling attackers to evade detection and blocking of their Command and Control (C&C) servers. Monitoring and blocking all such domains is generally challenging. Additionally, domains generated by DGAs are often created in real-time, making it difficult to detect and block them promptly. Therefore, effective methods for detecting these malicious domains used in cyberattacks are highly desired.

Previous research has shown that generating features from DGA-specific information, such as statistical information of domain strings, publicly available information on the Internet, and DNS query and response information, and then using machine learning to determine whether domains are malicious or not is effective. When evaluating proposed methods in existing research, datasets of malicious and benign domains are constructed from domains listed on blacklists [1]–[5] and domains with high user access, such as those in the Alexa DB [6].

The authors of [7] obtained DNS access logs of 45,000 benign domains from Rapid7 Labs [8] and DNS access logs of 45,000 malicious domains from the publicly available list of the SANS Internet Storm Center [9], and constructed and provided a dataset for malicious domain detection. In this study, we use the dataset from [7] to generate features based on the statistical information of domain strings and publicly available information on the Internet, and evaluate the detection performance using various classifiers.

II. Methods of Malicious Domain Generation

A. Domain Generation Algorithm (DGA)

A Domain Generation Algorithm (DGA) is an algorithm used by attackers to dynamically generate a large number of domain names, among which a Command & Control (C&C) server is selected for communication. This method is used to evade detection and blocking of C&C servers. Conficker [10] is a worm that was first discovered in 2008. It is said to generate thousands of different domain names daily using DGA, attempting to communicate with its C&C server through these domains. The generated domains are characterized by long random strings.

Zeus [11], a trojan malware discovered around 2007, primarily aims to steal online banking credentials. Zeus also employs DGA to generate different domain names daily for communicating with its C&C server. The DGA used by Zeus generates random strings combining alphabets and numbers. Because DGAs generate a large number of domains, it is challenging to monitor and block all of them. Additionally, DGA-generated domains are often created in real-time, making it difficult to detect and block them promptly.

B. Brand Imitation (Typosquatting)

Typosquatting is a technique that involves registering fake domain names by exploiting slight misspellings or typographical errors of legitimate brand names or domain names. The purpose is to deceive users into entering what appears to be a legitimate URL, thereby directing them to phishing pages. For example, if there is a legitimate domain named "cyber-attack.com," similar domains such as "cybar-attack.com" or "cyberattack.com" may be generated. Since numerous different misspellings or typographical errors can be considered for a single legitimate domain name, listing and monitoring all of them is challenging. Additionally, typosquatting domains closely resemble legitimate domains, making detection difficult both manually and by automated systems.

C. Homograph Attack

A homograph attack is a technique that involves using visually similar characters or symbols to create domain names that appear nearly identical to legitimate ones. For example, if there is a legitimate domain named "cyber-attack.co.jp," a spoofed domain might use characters that look similar, such as "cyber-attacк.co.jp" or "cyber-attack.co.jp."

In these examples, the legitimate domain uses Latin characters, whereas the spoofed domains use Cyrillic characters like "к" and "а."

Homograph attacks generate domains that closely resemble legitimate ones, making it difficult for users and security systems to distinguish them based on appearance alone. Additionally, homograph attacks can leverage multiple languages, utilizing various character sets, making it challenging to monitor all possible character sets comprehensively.

D. Bulk Registration

Bulk registration is a technique in which attackers use scripts or bots to purchase large quantities of expired or unused domain names, registering hundreds to thousands of domain names at once. Using these bulk-registered domains, attackers can create numerous sites for distributing malware or hosting phishing pages. The attackers make detection difficult by frequently changing domain names and using them for short periods before discarding them.

E. Subdomain Abuse

There is a method of exploitation that involves using subdomains. This technique hosts malicious content as a subdomain of a trusted parent domain. As a result, users are more likely to mistakenly trust the subdomain based on the reputation of the parent domain. For instance, if there is a trusted parent domain "example.com," an attacker might create a subdomain such as "malicious.example.com." Users may then perceive this subdomain as safe, based on the trustworthiness of "example.com," and be directed to phishing or malware sites. Attackers often generate subdomains dynamically and use them for short periods, making detection difficult. Additionally, adding the entire trusted parent domain to a blacklist is impractical, making blacklist-based detection challenging.

III. Related Works: Malicious Domain Detection

A. Messabi et al. [12]

Messabi et al. [12] proposed a method for detecting malicious domains using features based on domain names, TLDs, and tokens, employing a decision tree classifier. They collected 5,000 malicious domains from the Malware Domain List [1] and 4,400 benign domains from Alexa DB [6] and Google search engine. Their proposed method achieved an accuracy of 0.775 and a recall of 0.788.

B. DomainWatcher: Zhang et al. [13]

Zhang et al. [13] proposed a method called DomainWatcher for detecting malicious domains based on lexical features, mimicry features that evaluate the similarity to popular domain names using Levenshtein distance, and bigram features that assess similarity to popular domain names using bigrams. They collected 2,000 malicious domains from PhishTank [2], DNS-BH [1], and 5,000 benign domains from Alexa DB [6]. DomainWatcher achieved a recall of 0.963.

C. DomainProfiler: Chiba et al. [14]

Chiba et al. [14] proposed a system called DomainProfiler that detects newly registered malicious domains using a Random Forest classifier based on statistical features, DNS-based features, and temporal change pattern features. They used a dataset comprising 83,760 malicious domains collected from the hpHosts list [4] and 89,739 benign domains from Alexa DB [6]. The system achieved an accuracy of 0.990 and a recall of 0.975.

D. Cho et al. [15]

Cho et al. [15] collected malicious domains from PhishTank [2] and Malware Domain List [3], and benign domains from Alexa DB [6] and Public-domain-list [16], using these datasets for their study. They proposed a method for detecting malicious domains using a Random Forest classifier based on lexical features, ranking-based features, and DNS query features. The method achieved an accuracy of 0.969 and a recall of 0.968.

E. Kazato et al. [17]

Kazato et al. [17] proposed a method for detecting malicious domains using a graph convolutional network based on domain-IP relationships, domain owner information, and AS numbers to construct a domain relationship graph. They collected 10,000 malicious domains from Abuse.chZeuSTracker [5] and DNS-BH [1], and 10,000 benign domains from Alexa DB [6], using these datasets for their study. The method achieved an accuracy of 0.863 and a recall of 0.833.

F. Summary

The study by Messabi et al. [12] achieved a recall of 0.778 using only a few types of information derived from domain names. In Zhang et al.'s work [13], the use of different feature sets resulted in a recall of 0.963 through cross-validation. The studies by Chiba et al. [14] and Cho et al. [15] achieved high-accuracy detection of malicious domains by using features obtained from external data sources in addition to information retrievable from domain names. Kazato et al.'s research [17] achieved relatively high accuracy by relying primarily on information obtained from external sources rather than string features. Combining these string features with external information is expected to further improve accuracy.

In this paper, we evaluate a model for detecting malicious domains using an approach that employs statistical features of domain name strings and publicly available information from external data sources such as WHOIS [18] and Alexa DB [6]. This approach uses a publicly available dataset, which differs from the datasets commonly used in previous studies, to evaluate the performance of the detection model.

IV. Related Works: DNS dataset for malicious domains detection

The authors of [7] constructed a dataset by obtaining DNS access logs for 45,000 benign domains from Rapid7 Labs [8] and DNS access logs for 45,000 malicious domains from the public list of the SANS Internet Storm Center [9]. Their dataset includes 90,000 domain names and 34 features, including the characteristics of domain name strings, information obtained from WHOIS [18], whether they are included in AlexaDB [6], whether they are listed on public blacklists, and information obtained from DNS queries and responses. This study focuses on the following three categories:

Statistical Information of Strings
: Benign domain names are generally designed to be memorable and easy to use. Therefore, features directly obtainable from domain names are used as statistical information of strings.

Public Information
: Malicious domains tend not to publish or falsify registration information, while benign domains are considered to be older and have longer durations. Therefore, features obtained from external data sources such as WHOIS [18] and AlexaDB [6] are used as public information.

DNS Query and Response Information
: Features obtained from the response information when sending DNS query packets are used as DNS query and response information. It should be noted that the dataset from [7] is required to anonymize some columns and not to disclose personal data in accordance with GDPR rules, and as a result, some features in the dataset have values that cannot be identified.

V. Pre-processing and Feature Engineering

In this study, we use the dataset from [7] to examine how each of the following contributes to the detection of malicious domains: domain name string characteristics, information obtained from WHOIS [18], inclusion in AlexaDB [6], presence on public blacklists, and information obtained from DNS access logs.

The generation of features is based on the following pre-processing steps on the dataset from [7]:

Step 1. Remove anonymized features (domain names, IP addresses) whose original values are unknown.

Step 2. Remove features with a significant amount of missing data (registration country code).

Step 3. Convert the following categorical variables into dummy variables using one-hot encoding:
- DNS record type
- Country code
- Domain creation date
- Domain last update date
- ASN
- HTTP response code
- Domain registration organization name
- TLD

Step 4. Group the following dummy variables to reduce model complexity and prevent overfitting: "Country code," "ASN," "Domain registration organization name," "TLD."

The registration country code, which had about 90% missing data, was removed in Step 2. When converting categorical variables into dummy variables in Step 3, those with many categories were grouped in Step 4 to reduce model complexity and prevent overfitting. Specifically, the country code was grouped into "US," "Other," and "null" because [7] indicates that most IPs were active in the United States and many of them were malicious. The ASN was grouped into "26228," "Other," and "null" based on [7], which states that many domains had ASN 26228. The domain registration organization name was grouped into "null" or "not null" based on the assumption that legitimate organizations typically do not hide their registration organization name when registering domains. The TLD was grouped into "com" and "other than com" because [7] indicates that many domains had a "com" TLD.

VI. Evaluation

A. Objectives and Methods

The objective of this study is to evaluate the detection of malicious domains using machine learning with features derived from the statistical characteristics of domain strings, publicly available information, and DNS query-response information. This evaluation is performed using a publicly available dataset that is different from those used in previous studies.

Following the preprocessing steps described in section V., the dataset is divided into training and test data based on the domain creation date. Specifically, domains created within the last year are used as test data. Due to the small amount of

Table I
CONFUSION MATRIX

		Predicted	
		Malicious	Benign
Actual	Malicious	TP (True Positive)	FN (True Positive)
	Benign	FP (False Positive)	TN (True Negative)

data within the last year, resulting in a limited number of test samples, we conducted 5-fold cross-validation with fixed test data. In detail, the training data is divided into five parts, with four sets used for training and evaluation.

Tree-based learning algorithms are computationally efficient and suitable for large datasets. We evaluate three tree-based learning algorithms: Decision Tree (DT), Random Forest (RF), and XGBoost (XGB). Basic classification algorithms such as logistic regression were also evaluated in the same manner, but their performance was not high, so they are omitted.

B. Results

We conducted experiments using five different combinations of three types of features: (a) using only statistical information of strings, (b) using only public information, (c) using only DNS query-response information, (d) using statistical information of strings and public information ((a) + (b)), and (e) using statistical information of strings and DNS query-response information ((a) + (c)). The models were evaluated based on the confusion matrix shown in Table I, and Precision, Recall, and F1-score were calculated as follows:

$$\begin{aligned} \text{Precision} &= \text{TP}/(\text{TP}+\text{FP}) \\ \text{Recall} &= \text{TP}/(\text{TP}+\text{FN}) \\ \text{F1-score} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

The experimental results are shown in Table II. Each evaluation metric is rounded to three decimal places. The experimental results indicate that all three classifiers (DT, RF, XGB) achieved a recall of at least 0.940 and an F1-score of at least 0.892. The method using both statistical information of strings and DNS query-response information achieved the highest F1-score across all classifiers. The highest F1-score among all the experimental results was 0.984, achieved by both the DT and XGB classifiers using the method that combines statistical information of strings and DNS query-response information.

VII. COMPARISONS

We compare our study [12], which uses only statistical features of domain strings, and [15], which uses those features plus public information and DNS query-response features. These two methods were not re-evaluated in our study; their performance metrics are transcribed from the respective papers.

Table III shows the features and detection performance used in related studies and this study. The RF classifier using only statistical features in our study achieved higher values in Precision, Recall, and F1-score compared to [12]. As shown in Table III, although both studies use statistical features of strings, our study uses a greater number of features. Features not included in [12] (e.g., ratio of digits, ratio of consonants) were found to be highly important in the feature importance rankings of our XGB classifier, as shown in Table IV.

As shown in Table III, our RF classifier using statistical features of strings and public information achieved higher Recall and F1-score compared to [15]. Reference [15] includes features derived from DNS query-response results (Nos. 48 to 57 in Table III), suggesting that information from DNS queries and responses is effective for detecting malicious domains.

The XGB classifier in our study, using statistical features of strings and DNS query-response information, achieved higher values in Precision, Recall, and F1-score compared to [15].

Table IV
STRING FEATURES OF HIGH IMPORTANCE IN XGBOOST

No.	Features	Importance
7	Numeric ratio	0.633
3	Maximum number of Numeric sequences	0.260
9	Consonant ratio	0.050
12	Strange characters	0.023

VIII. CONCLUSION

Previous research on malicious domain detection has demonstrated that generating DGA-specific features such as statistical information of domain strings, publicly available information on the internet, and DNS query and response information, and then using machine learning to determine whether a domain is malicious, is effective. In evaluating their proposed methods, existing studies typically construct datasets of malicious and benign domains from blacklists and domains with high user access, such as those in the Alexa DB, and evaluate their performance. This methodology makes it difficult to compare performance with existing methods, highlighting the need for evaluation using open datasets.

In this study, we focus on a publicly available dataset for malicious domain detection [7] and calculate performance metrics using basic classifiers, comparing these with the features and performance used in past research. When using only statistical features of domain strings, the Random Forest and XGBoost classifiers achieved a recall of 0.946 and an F1-score of 0.968. The Random Forest classifier using statistical features of domain strings and public information achieved a recall of 0.940 and an F1-score of 0.968. The XGBoost classifier using statistical features of domain strings and DNS query-response information achieved a recall of 0.983 and an F1-score of 0.984.

The publicly available dataset from [7] has different features from those used in existing research datasets, and its usefulness has been confirmed. Through the re-evaluation of existing methods, the generalization ability, robustness, reproducibility, and practicality of malicious domain classifiers can be confirmed, while also allowing for the re-evaluation of the

Table II
PERFORMANCE OF DECISION TREE (DT), RANDOM FOREST (RF), AND XGBOOST (XGB) FOR FIVE DIFFERENT COMBINATIONS OF FEATURES. NOTE THAT CLASSIFIERS A, B, AND C ARE REFERENCED IN TABLE III

Features			Classifers	Precision	Recall	F1-score	Hyperparameters
String	Public Info.	DNS logs					
✓			DT	0.990	0.940	0.964	max_depth:10, min_samples_split:4
			RF (Classifier A)	**0.991**	**0.946**	**0.968**	max_depth:10 min_samples_split:10, n_estimators:150
			XGB	0.990	**0.946**	**0.968**	gamma:0, max_depth:3, n_estimators:50
	✓		DT	**0.837**	0.955	0.892	max_depth:10, min_samples_split:8
			RF	0.829	**0.985**	**0.901**	max_depth:10, min_samples_split:10, n_estimators:200
			XGB	0.831	0.975	0.898	gamma:0, max_depth:3, n_estimators:150
		✓	DT	**0.962**	0.991	0.976	max_depth:10, min_samples_split:2
			RF	**0.962**	0.991	0.976	max_depth:10, min_samples_split:2, n_estimators:50
			XGB	**0.962**	**1.000**	**0.980**	gamma:0.3, max_depth:3, n_estimators:50
✓	✓		DT	0.977	**0.952**	0.965	max_depth:10, min_samples_split:2
			RF (Classifier B)	**0.998**	0.940	**0.968**	max_depth:10, min_samples_split:6, n_estimators:200
			XGB	0.985	0.946	0.965	gamma:0.5, max_depth:3, n_estimators:50
✓		✓	DT	**0.988**	0.980	**0.984**	max_depth:10, min_samples_split:6
			RF	0.981	**0.986**	0.982	max_depth:20, min_samples_split:4, n_estimators:50
			XGB (Classifier C)	0.986	0.983	**0.984**	gamma:0, max_depth:3, n_estimators:150

importance of specific features and the discovery of new features.

ACKNOWLEDGMENT

This work was partially supported by JST K Program Japan Grant Number JPMJKP24K1.

REFERENCES

[1] "Malware Domains. [n.d.]. Malware Prevention through DNS Redirection," https://riskanalytics.com/community/
[2] "Join the fight against phishing," https://www.phishtank.com/
[3] "Malware Domain List," http://www.malwaredomainlist.com/
[4] "Malwarebytes. hpHosts," http://www.hosts-file.net/
[5] "Zeus Tracker Abuse.ch," https://zeustracker.abuse.ch/
[6] "Amazon, AWS Alexa Top 1M, "http://s3.amazonaws.com/alexa-static/top-1m.csv.zip
[7] Marques, C., Malta, S. & Magalhães, J. DNS dataset for malicious domains detection. *Data In Brief*. **38** pp. 107342 (2021)
[8] Rapid7 Labs, "Forward DNS, " https://opendata.rapid7.com/sonar.fdns_v2
[9] "Suspicious Domain List," http://web.archive.org/web/20200503151842/https://www.dshield.org/feeds/suspiciousdomains_Low.txt
[10] J. Brooks, "Conficker: What It IS, How to Stop It and Why You May Already Be Protected," https://www.eweek.com/security/conficker-what-it-is-how-to-stop-it-and-why-you-may-already-be-protected/
[11] M. Higgins, "Zeus malware: Its history and how it works | NordVPN, " https://nordvpn.com/ja/blog/zeus-virus/
[12] Messabi, K., Aldwairi, M., Yousif, A., Thoban, A. & Belqasmi, F. Malware detection using DNS records and domain name features. *Proceedings Of The 2nd International Conference On Future Networks And Distributed Systems*. (2018)
[13] Zhang, P., Liu, T., Zhang, Y., Ya, J., Shi, J. & Wang, Y. Domain Watcher: Detecting Malicious Domains Based on Local and Global Textual Features. *Procedia Computer Science*. **108** pp. 2408-2412 (2017), International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland
[14] Chiba, D., Yagi, T., Akiyama, M., Shibahara, T., Yada, T., Mori, T. & Goto, S. DomainProfiler: Discovering Domain Names Abused in Future. *2016 46th Annual IEEE/IFIP International Conference On Dependable Systems And Networks (DSN)*. pp. 491-502 (2016)
[15] Xuan, C. Malicious domain detection based on DNS query using Machine Learning. *International Journal Of Emerging Trends In Engineering Research*. (2020)
[16] "Public-domain-list," https://github.com/opendns/public-domain-lists
[17] Kazato, Y., Nakagawa, Y. & Nakatani, Y. Improving Maliciousness Estimation of Indicator of Compromise Using Graph Convolutional Networks. *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. pp. 1-7 (2020)
[18] "IPWhois," https://pypi.org/project/ipwhois/

Table III
FEATURES AND DETECTION PERFORMANCE USED IN RELATED STUDIES AND THIS STUDY

		Classifier	Ref. [12]	Ref. [15]	Classifier A	Classifier B	Classifier C
		Precision	0.852	0.976	0.991	0.998	0.986
		Recall	0.699	0.968	0.946	0.940	0.983
		F1-score	0.768	0.972	0.968	0.968	0.984
Categories	No.	Features					
String Features	1	Domain length		✓	✓	✓	✓
	2	Maximum number of character sequences		✓			
	3	Maximum number of numeric sequences	✓	✓	✓	✓	✓
	4	Maximum number of consonant sequence			✓	✓	✓
	5	Maximum number of vowel sequence			✓	✓	✓
	6	Maximum number of special char sequence			✓	✓	✓
	7	Numeric ratio			✓	✓	✓
	8	English ratio	✓				
	9	Consonant ratio			✓	✓	✓
	10	Vowel ratio			✓	✓	✓
	11	Special char ratio			✓	✓	✓
	12	Strange characters			✓	✓	✓
	13	Entoropy			✓	✓	✓
	14	Number of dots	✓	✓			
	15	Number of hyphens	✓				
	16	Average number of characters between dots		✓			
	17	Maximum number of characters between dots		✓			
	18	Number of IP address in domain name		✓			
	19	Number of special characters		✓			
	20	Number of continuous numeric character		✓			
	21	number of characters	✓				
	22	Number of continuous digits		✓			
	23	Number of continuous letters		✓			
	24	TLD	✓		✓	✓	✓
	25	Maximum levenshtein ratio		✓			
	26	Inappropriate words presence	✓				
	27	Transfer-based words presence	✓				
	28	Brand name presence		✓			
Public Info.	29	Domain registration		✓			
	30	Domain creation date				✓	
	31	Domain last update date				✓	
	32	Common port active				✓	
	33	Country code				✓	
	34	Domain registration organization name				✓	
	35	ASN				✓	
	36	IP block list search				✓	
	37	Domain block list search				✓	
	38	number of subdomains				✓	
	39	Entropy of subdomains				✓	
	40	Included in Alexa DB				✓	
	41	Rank in Alexa host		✓			
	42	Rank in Alexa country		✓			
	43	Rank in Domcop		✓			
	44	HTTP response code		✓		✓	
	45	SSL certificate		✓			
	46	Age domain		✓			
DNS logs	47	DNS record type					✓
	48	Resolved IP count		✓			
	49	Number of distinct country		✓			
	50	Silent IP ratio		✓			
	51	Number of name servers		✓			
	52	Number of IP addresses of name servers		✓			
	53	Number of name server Country		✓			
	54	Number of mail exchange server		✓			
	55	Number of NS record		✓			
	56	Number of CName record		✓			
	57	MX DNS responce		✓			
	58	TXT DNS responce					✓
	59	Entropy of subdomains					✓
	60	Has sender policy framework					✓
	61	Has Domain keys identified email					✓
	62	Has Domain based massage authentication					✓