Detection of DGA-based Malware Communications from DoH Traffic Using Machine Learning Analysis

Rikima Mitsuhashi[1,2] , Yong Jin[3] , Katsuyoshi Iida[2] , Takahiro Shinagawa[1] , Yoshiaki Takai[2]
[1] *The University of Tokyo, Tokyo, Japan*
[2] *Hokkaido University, Sapporo, Japan*
[3] *Tokyo Institute of Technology, Tokyo, Japan*
E-mail: mitsuhashi@os.ecc.u-tokyo.ac.jp, yongj@gsic.titech.ac.jp, iida@iic.hokudai.ac.jp,
shina@ecc.u-tokyo.ac.jp, ytakai@iic.hokudai.ac.jp

***Abstract*—Encrypted domain name resolution can reduce the risk of privacy leakage for Internet users, but it may also prevent network administrators from detecting suspicious communications. Since operating systems supporting DNS over HTTPS (DoH) have increased in recent years, malware that uses Domain Generation Algorithm (DGA) can exploit it to hide the generated domain names. In this paper, we propose a system that detects DGA-based malware communications from DoH traffic. Based on the concept of hierarchical machine learning analysis, the proposed system classifies network traffic with Gradient Boosting Decision Tree (GBDT) and tree-ensemble models. The evaluation confirmed that the system was able to detect DoH traffic generated by PadCrypt, Sisron, Tinba, and Zloader with 99.12% accuracy. The results indicate that the system has the ability to detect different DGA-based malware communications from DoH traffic with sufficient accuracy to support network administrators.**

***Index Terms*—DNS over HTTPS (DoH), Hierarchical network traffic classification, Gradient Boosting Decision Tree (GBDT), Regularized Greedy Forest (RGF), Domain Generation Algorithm (DGA), DGA-based malware.**

I. Introduction

Encrypted domain name resolution has become increasingly used to protect the privacy of Internet users. There are two leading encryption methods. One is DNS over HTTPS (DoH), which encrypts DNS queries/responses over HTTPS on port 443, and the other is DNS over TLS (DoT), which encrypts DNS queries/responses over TLS on port 853. Currently, the DoH is more popular because of its faster deployment in web browsers and better port number compatibility with the existing firewalls. In recent years, operating systems supporting DoH have increased. For example, Windows 11, released in October 2021, provides DoH encryption named secure DNS client over HTTPS [1]. MacOS 11 and iOS 14, launched in September 2020, can use a DoH configuration referred to as NEDNSsettingsManager [2]. As for Linux, although DoH is not yet officially supported, it is possible to install DoH proxy software such as DNS-over-HTTPS [3], DNSCrypt [4], and doh-client [5].

Encrypted domain name resolution can reduce the risk of privacy leakage for Internet users, but it may also prevent network administrators from detecting suspicious communications. When client users in an organization choose a public DNS service that provides DoH connection interfaces, network administrators in the organization cannot use domain names for security monitoring. Since operating systems supporting DoH have increased recently, malware can exploit it to hide domain names in communications. Even under the circumstances, network administrators must maintain network security.

To detect malware communications from network traffic, we focus on malware that uses the Domain Generation Algorithm (DGA), an algorithm that automatically creates random domain names. The malware, called DGA-based malware, generates millions of domain names and sends them to domain name resolution servers to find command and control (C&C) servers somewhere on the Internet. Cyber attacks using DGA-based malware are hazardous, and thousands of organizations have been seriously compromised [6]. Unfortunately, DGA-based malware can exploit the latest operating systems supporting DoH to hide the generated domain names. In the literature, many approaches have been proposed to detect malware communications on the basis of malicious domain names [7]–[17]. However, these approaches are difficult to apply directly to encrypted traffic such as DoH because they assume that the domain names are in plain text.

In this paper, we propose a system that detects DGA-based malware communications from the DoH traffic. Based on the concept of the hierarchical machine learning analysis in Fig. 1, the proposed system filters DoH traffic from HTTPS traffic in the 1st stage, recognizes the suspicious DoH traffic in the 2nd stage, and detects communications generated by DGA-based malware in the 3rd stage. Each stage incorporates Gradient Boosting Decision Tree (GBDT) and tree-ensemble models. The specific models are XGBoost [18], LightGBM [19], CatBoost [20], and Regularized Greedy Forest (RGF) [21]. To the best of our knowledge, this is the first report of DGA-based malware detection from the DoH traffic.

The evaluation confirmed that parameter-tuned LightGBM in the 3rd stage detected DoH traffic generated by PadCrypt [22], Sisron [23], Tinba [24], [25], and Zloader [26] with 99.12% accuracy. In addition, parameter-tuned LightGBM in the 1st stage filtered DoH traffic with 99.92% accuracy, and parameter-tuned CatBoost in the 2nd stage recognized suspi-

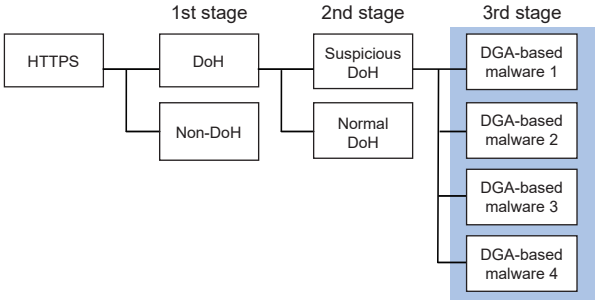

Fig. 1. Concept of hierarchical machine learning analysis.

cious DoH traffic with 99.97% accuracy. The effectiveness of the proposed system is its ability to classify different malware communications from DoH traffic with sufficient accuracy to support network administrators. Understanding the types of DGA-based malware allows them to improve the anti-malware solutions in the network.

The main contributions of this paper are as follows.

- We propose a system that detects DGA-based malware communications from DoH traffic.
- The proposed system based on hierarchical machine learning analysis can accurately classify encrypted network traffic.
- The proposed system leads to mitigating the threat of cyber attacks with DGA-based malware.

The remainder of this paper is organized as follows. Section II summarizes related research on DGA domain detection and DoH traffic classification. Section III proposes a system design based on hierarchical machine learning analysis. Section IV describes experimental evaluation detecting DGA-based malware communications. Section V concludes the paper.

II. Related Research

A. DGA domain detection

DGA is an algorithm that automatically creates random domain names, and the DGA-based malware sends them to domain name resolution servers, as described in the prior section. Since domain name resolution is one of the primitive services on the Internet, previous studies have reported countermeasures against attacks that exploit the DNS protocol [7]–[11].

In recent years, S. Ajmera and T. Pattanshet [12] have reported that machine learning and deep learning research for DGA domain detection has increased. For example, H. Suryotrisongko et al. [13] proposed a model to detect DGA-based traffic based on statistical features based on entropy using Shannon's function, domain name character length, and Alexa reputation score. Their experimental results showed random forest model achieved 96.3% classification accuracy on the datasets comprising 55 DGA families. M.A. Ayub et al. [14] extracted features from the Bigram and Word2Vec models to detect malicious DGA domains. They applied the features for text processing in combination with machine learning and deep learning techniques to analyze 84 different malware families. The experimental results showed that Artificial Neural Network (ANN) method using the Bigram model performed the best with an accuracy of 99.8% in detecting benign and malicious domains and 93.58% in classifying domains belonging to specific malware families. Y. Zhang et al. [15] implemented DGA family clustering to identify DGA families. They investigated 22 different DGA families to find a practical approach to DGA family classification. The experiment identified the six DGA families with the k-nearest neighbor algorithm. R.R. Curtin et al. [16] devised the smashword score, which measures how much a DGA family looks like English words. They used a model that combines recurrent neural network architecture and domain registration side information. The experiments detected DGA families such as matsnu, suppobox, rovnix, and others. D. Plohman et al. [17] performed a measurement study of the DGA by analyzing 43 DGA-based malware families and variants. They presented a taxonomy for DGAs to characterize and compare their properties. They confirmed that pre-computing future DGA domains could identify corresponding malware families and related campaigns.

Although these studies effectively detect DGA domain names, applying their proposed methods directly to the DoH traffic is difficult because they assume that the DGA domain names are in plain text.

B. DoH Traffic Classification

The DoH protocol is a method of encrypting domain names, as mentioned above. On the other hand, many studies using machine learning have reported that it can recognize certain types of DoH traffic, including web and tunneling communications.

As for web communications, L. Csikor et al. [27] used machine learning to classify web or DoH on HTTPS traffic, achieving a classification accuracy of 97.4% in a closed environment and 90% in an open environment. The dataset in the experiments included thousands of domains in Alexa's list of top-ranked websites. D. Vekshin et al. [28] recognized DoH traffic from HTTPS traffic with 99.6% accuracy. They classified DoH client programs, including Chrome, Firefox, and Cloudflared [29], with 99.9% accuracy. Their best machine learning model was the Ada-boosted decision tree in both experiments. The dataset in their experiments consisted of 1 million domains served by Alexa top sites [30].

Regarding tunneling communications, M. MontazeriShatoori et al. [31] filtered DoH traffic from HTTPS traffic and then recognized malicious DoH traffic from DoH traffic. They use the malicious label for the DoH tunneling traffic, which malicious users can use to create covert channels. In their experiments using Random Forest on the CIRA-CIC-DoHBrw-2020 dataset [32], filtering obtained an F-score of 99.3%, and recognition resulted in an F-score of 99.9%. R. Mitsuhashi et al. [33] implemented a traffic classification system and identified malicious DNS tunnel tools on DoH traffic. The experimental results showed that the classification accuracy of the CIRA-CIC-DoHBrw-2020 dataset [32] was 97.22%. Y.

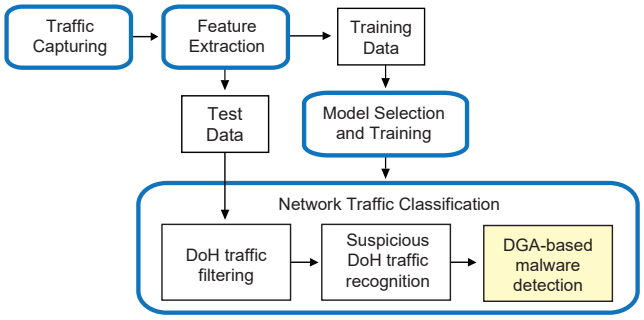

Fig. 2. Overview of proposed system to detect DGA-based malware.

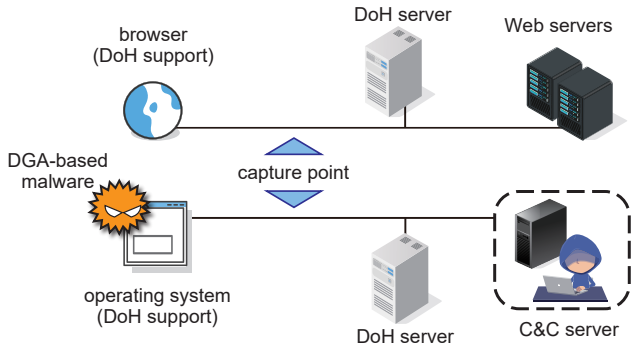

Fig. 3. Network connections and capture points.

Khodjaeva et al. [34] recognized malicious DNS tunnels by investigating the entropy of a flow. Their evaluations confirmed that the C4.5 Decision Tree classifier achieved an F-measure of 99.7% on training datasets when statistical features obtained by Tranalyzer2 flow exporter are augmented with the entropy of a flow over the first four packets.

Previous DoH traffic classification studies recognize web and tunneling communications with high classification accuracy. Still, there have been no studies to detect DGA-based malware communications from DoH traffic as far as we know. Therefore, in this paper, we proposed a system that detects DGA-based malware communications from DoH traffic using machine learning analysis.

III. System Design

As mentioned in the previous section, there have been no studies to detect DGA-based malware communications from DoH traffic as far as we know. In this section, we propose a system design based on hierarchical machine learning analysis for DGA-based malware detection.

A. Overview

We illustrate an overview of the proposed system to detect DGA-based malware in Fig. 2. The proposed system includes four blocks: traffic capturing, feature extraction, model selection and training, and network traffic classification. The network traffic classification block consists of three components: DoH traffic filtering, suspicious DoH traffic recognition, and DGA-based malware detection. We explain each block below.

B. Traffic Capturing and Feature extraction

The proposed system uses HTTPS traffic on port number 443 as input. Fig. 3 shows the Network connections and capture points. The browser supporting DoH connects to a DoH server for domain name resolution (Normal DoH). It also accesses web servers to retrieve web content (Non-DoH). The DGA-based malware connects to the DoH server through the operating system supporting DoH for domain name resolution (Suspicious DoH). This connection returns an NXDOMAIN message to the DGA-based malware in order to keep a secure experimental environment. Therefore, DGA-based malware cannot connect to the C&C server. To extract traffic features, we use DoHlyzer [35] to automatically obtain 28 statistical traffic features from captured HTTPS traffic data at intervals of up to approximately 2 minutes, as shown in Table I.

TABLE I
List of statistical traffic features.

1	Number of Flow Bytes Sent	17	Standard Deviation of Packet Time
2	Rate of Flow Bytes Sent	18	Coefficient of Variation of Packet Time
3	Number of Flow Bytes Received	19	Skew from Median Packet Time
4	Rate of Flow Bytes Received	20	Skew from Mode Packet Time
5	Mean Packet Length	21	Mean Request/response Time Difference
6	Median Packet Length	22	Median Request/response Time Difference
7	Mode Packet Length	23	Mode Request/response Time Difference
8	Variance of Packet Length	24	Variance of Request/response Time Difference
9	Standard Deviation of Packet Length	25	Standard Deviation of Request/response Time Difference
10	Coefficient of Variation of Packet Length	26	Coefficient of Variation of Request/response Time Difference
11	Skew from Median Packet Length	27	Skew from Median Request/response Time Difference
12	Skew from Mode Packet Length	28	Skew from Mode Request/response Time Difference
13	Mean Packet Time		
14	Median Packet Time		
15	Mode Packet Time		
16	Variance of Packet Time		

C. Model Selection and Training

This subsection explains how to select the machine learning models for the hierarchical machine learning analysis. The proposed system incorporates GBDT and tree-ensemble models. The specific models are XGBoost, LightGBM, CatBoost, and RGF. S. R et al. [36] describe that GBDT is far more flexible and needs a shorter training time than other current machine learning algorithms. Additionally, according to the RGF websites [37], it can deliver better results than GBDT on a number of datasets.

For high classification accuracy, it is important not only to use high-performance machine learning models but also to tune the parameters in order to fit the dataset. Fig. 4 shows the model selection process. First, the parameter-tuned models and training data are used for a grid search. Then, the best accuracy model is selected based on the grid search results. Finally, the model is trained on the training data to create a classifier. The process is done for three stages of network traffic classification, resulting in three classifiers.

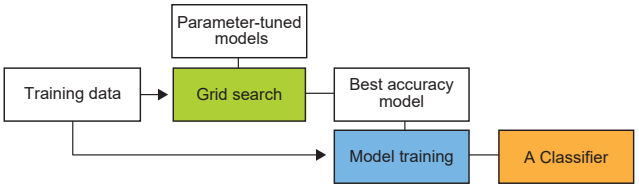

Fig. 4. Model selection and training process.

TABLE II
GRID SEARCH PARAMETERS
(UNDERLINE SHOWS DEFAULT PARAMETER).

XGBoost	LightGBM	CatBoost	RGF
max_depth: 2, 4, 6, 8, 10, 12, 14	num_leaves: 3, 7 ,15, 31, 63, 127, 255, 511, 1023	max_depth: 2, 6, 10, 14, 16	max_leaf: 500, 1000, 2000, 4000, 8000, 16000
max_bin: 64, 128, 256, 512, 1024, 2048, 4096, 8192	max_bin: 64, 127, 255, 511, 1023, 2047, 4095	l2_leaf_reg: 1, 3, 5, 7, 9	l2: 1.0, 0.1, 0.01

The types of parameters tuned in each model are listed in Table II. Their effectiveness in improving classification accuracy is described in the documentation for each model. The parameter values are changed incrementally from the default to obtain 162 models: 56 models for XGBoost, 63 models for LightGBM, 25 models for CatBoost, and 18 models for RGF.

When training machine learning models, we need to be careful about overfitting. Overfitting is a problem in which a model closely related to a particular data set fails to classify the additional data correctly. If the model is overfitted to the training data by parameter tuning, it cannot classify the test data with sufficient accuracy. Therefore, verification of overfitting requires a comparison of classification accuracy for both training and test data.

D. Network Traffic Classification

The proposed system classifies the network traffic through three stages to detect communication generated by DGA-based malware. The process in Section III-C selects the classifier used in each stage. Fig. 5 illustrates the input and output data of the three-stage classifiers. When the HTTPS traffic is input, the 1st stage classifier filters the DoH traffic. Then, the 2nd stage classifier recognizes the suspicious DoH traffic from the DoH traffic. Finally, the 3rd stage classifier detects communications generated by different DGA-based malware instances from the suspicious DoH traffic.

IV. EXPERIMENTAL EVALUATION

The former section presented the proposed system design and illustrated the three-stage classification process. In this section, we describe the experimental evaluation detecting DGA-based malware communications.

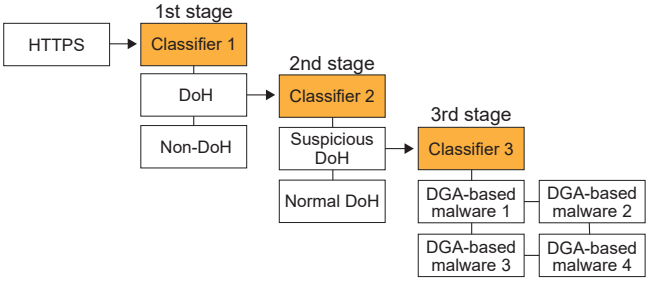

Fig. 5. Network traffic classification process.

TABLE III
LABELS AND TRAFFIC FLOWS FOR EVALUATION.

	Labels	Traffic flows
1st stage (HTTPS)	Non-DoH	897493
	DoH	24019
2nd stage (DoH)	Normal DoH	19807
	Suspicious DoH	4212
3rd stage (DGA-based malware)	PadCrypt	840
	Sisron	744
	Tinba	1808
	Zloader	820

A. Implementation

We implemented the proposed system using XGBoost 1.4.2, LightGBM 3.3.0, CatBoost 1.0.0, and RGF 3.11.0 on Ubuntu 20.04 and python 3.8.10.

B. Dataset

The number of labels and traffic flows for the evaluation is shown in Table III. We obtained non-DoH in the 1st and normal DoH in the 2nd stages from the CIRA-CICDoHBrw-2020 dataset (l1-nondoh.csv, l2-benign.csv). PadCrypt, Sisron, Tinba, and Zloader in the 3rd stage were generated by real DGA-based malware instances obtained from VirusShare [38]. They are intended to exploit Windows 11 as zero-day malware. Note that the small number of suspicious DoH compared to normal DoH in the 2nd stage represents a situation where malware communications are mixed in with user communications. To verify the numerical results by the readers, we publish the labels and traffic flows in the 3rd stage as the DoH-DGA-Malware-Traffic-HKD dataset [39].

C. Evaluation Metrics

We used accuracy, precision, recall, and F-score as evaluation metrics for classifications. For multi-class classification in the 3rd stage, the macro-average of each metric was applied. We also used stratified 10-fold cross-validation to evaluate the performance, splitting the training and test data into a 9:1 ratio.

D. Selected models

The classifiers at each stage were selected by comparing the accuracy of grid search on training data. The results in Table IV show that the 1st stage was parameter-tuned

TABLE IV
BEST ACCURACY AND PARAMETERS OBTAINED BY GRID SEARCH.

		1st stage	2nd stage	3rd stage
XGBoost	accuracy	0.9977	0.9995	0.9909
	max_depth	10	2	8
	max_bin	4096	64	128
LightGBM	accuracy	0.9991	0.9994	0.9923
	num_leaves	511	15	15
	max_bin	511	64	2047
CatBoost	accuracy	0.9974	0.9996	0.9915
	max_depth	16	14	10
	l2_leaf_reg	9	1	7
RGF	accuracy	0.9983	0.9994	0.9907
	max_leaf	16000	16000	4000
	l2	0.01	0.01	0.01

TABLE V
RESULTS OF DGA-BASED MALWARE COMMUNICATION DETECTION.

	Classifiers	Accuracy	Precision	Recall	F-score
1st stage	LightGBM	0.9992	0.9993	0.9998	0.9996
2nd stage	CatBoost	0.9997	0.9995	0.9992	0.9994
3rd stage	LightGBM	0.9912	0.9894	0.9897	0.9894

LightGBM, the 2nd stage was parameter-tuned CatBoost, and the 3rd stage was parameter-tuned LightGBM. The 1st and 3rd stages selected the same model, but the parameter values were quite different.

E. DGA-based malware communication detection

The results of DGA-based malware communication detection on test data using the classifiers selected in Section IV-D are shown in Table V. The LightGBM in the 3rd stage detected communications generated by four DGA-based malware instances, including PadCrypt, Sisron, Tinba, and Zloader, with 99.12% accuracy and 98.94% F-score. The LightGBM in the 1st stage filtered DoH traffic with 99.92% accuracy and 99.96% F-score. The Catboost in the 2nd stage recognized Suspicious DoH traffic with 99.97% accuracy and 99.94% F-score. The minimal difference between accuracy and F-score in each stage represents that there are no classes with extremely low classification accuracy.

Furthermore, the classification accuracies for each stage in Table IV and Table V were 99.91% and 99.92% for the 1st stage, 99.96% and 99.97% for the 2nd stage, and 99.23% and 99.12% for the 3rd stage. The nearly equal accuracy for each stage between training data and test data indicates no overfitting due to parameter tuning.

Overall, the results reveal that the proposed system has the ability to detect DoH communications of different DGA-based malware instances with sufficient accuracy to support network administrators. Understanding the types of DGA-based malware allows them to improve the anti-malware solutions in the network.

TABLE VI
MOST IMPORTANT FEATURES IN THE PROPOSED SYSTEM.

	Important features	Values
1st stage	Median Request/response Time Difference	3393
	Number of Flow Bytes Sent	2992
2nd stage	Median Request/response Time Difference	28.43
	Mode Request/response Time Difference	15.32
3rd stage	Number of Flow Bytes Sent	672
	Variance of Packet Length	431

F. Discussion of Important Features

To analyze the background behind the high classification accuracy achieved by the three-stage classifiers, the important features they used are listed in Table VI. The important features and values are obtained from the "feature_importances_" attribute of each classifier. In the 1st stage, the classifier considered "Median Request/response Time Difference" as the most important feature, which refers to the time it takes a client from sending a request to receiving a response. The average value of that feature over the dataset was 0.0591 for the non-DoH traffic and 0.0141 for the DoH traffic. The difference suggests that the response time of the DoH server tends to be faster than that of the Web server, depending on network distance or server load.

In the 2nd stage, as in the 1st stage, the classifier regarded "Median Request/response Time Difference" as the most important feature. The average value of that feature over the dataset was 0.0166 for normal DoH traffic and 0.0025 for suspicious DoH traffic. We found that the response time to DGA-based malware was faster than the response time to browsers because the DoH server almost always returns NXDOMAIN messages to DGA-based malware, as observed in many cases. In contrast, if DGA-based malware acquires the A record of C&C servers within a few minutes, The proposed system may not detect them with the expected high accuracy due to the lack of traffic flows to extract statistical features.

In the 3rd stage, the classifier ranked "Number of Flow Bytes Sent" as the most important feature, representing the total amount of data sent over a period. The average value of that feature over the dataset was 70050 for PadCrypt, 16543 for Sisron, 235890 for Tinba, and 6898 for Zloader. The difference in their values depends on how often each DGA-based malware sends domain name resolution queries to the DoH server. Even if malicious developers modify the behavior of existing DGA-based malware instances, it is expected to detect them through relearning. Detection of modified DGA-based malware instances, called variants, is future work.

V. CONCLUSION

DoH protocol has been standardized to protect the privacy of Internet users. However, it may prevent network administrators from detecting suspicious communications because DGA-based malware can exploit it to hide generated domain

names. The paper proposed a system that detects DGA-based malware communications from DoH traffic. We implemented the proposed system based on hierarchical machine learning analysis and evaluated its performance.

In the evaluation, the system detected DoH traffic generated by PadCrypt, Sisron, Tinba, and Zloader with 99.12% accuracy. The results indicate that the system has the ability to detect different DGA-based malware communications from DoH traffic with sufficient accuracy to support network administrators. Understanding the types of DGA-based malware allows them to improve the anti-malware solutions in the network. Moreover, we presented the important features to discuss the backgrounds behind the high classification accuracy. To verify the numerical results by the readers, we publish the dataset of DoH traffic flows generated by real malware instances. Future work includes the detection of DGA-based malware variants.

VI. Acknowledgments

This work was partially supported by JSPS KAKENHI (Grants-in-Aid for Scientific Research) Grant Number 19K20254.

References

[1] "Secure DNS Client over HTTPS (DoH)," https://docs.microsoft.com/en-us/windows-server/networking/dns/doh-client-support.

[2] "DNS on IOS v14 in Apple Developer Forums," https://developer.apple.com/forums/thread/663371.

[3] "DNS-over-HTTPS," https://github.com/m13253/dns-over-https.

[4] "DNSCrypt," https://github.com/DNSCrypt.

[5] "doh-client," https://docs.rs/crate/doh-client/1.1.5.

[6] "Sunburst: Supply Chain Attack Targets SolarWinds Users," https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/sunburst-supply-chain-attack-solarwinds.

[7] H. Ichise, Y. Jin, and K. Iida, "Analysis of DNS TXT Record Usage and Consideration of Botnet Communication Detection," *IEICE Transactions on Communications*, vol. E101, no. 1, pp. 70–79, 2018.

[8] H. Ichise, Y. Jin, K. Iida, and Y. Takai, "NS record History Based Abnormal DNS traffic Detection Considering Adaptive Botnet Communication Blocking," *IPSJ Journal of Information Processing*, vol. 28, pp. 112–122, 2020.

[9] Y. Iuchi, Y. Jin, H. Ichise, K. Iida, and Y. Takai, "Detection and Blocking of DGA-based Bot Infected Computers by Monitoring NXDOMAIN Responses," in *Proceedings of 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2020, pp. 82–87.

[10] H. Ichise, Y. Jin, and K. Iida, "Policy-based detection and blocking system for abnormal direct outbound DNS queries using RPZ," *Proceedings of International Conference on Future Computer and Communication (ICFCC 2022)*, 2022.

[11] J. Y. Lee, J. Y. Chang, and E. G. Im, "DGA-Based Malware Detection Using DNS Traffic Analysis," in *Proceedings of the Conference on Research in Adaptive and Convergent Systems*, ser. RACS '19, 2019, p. 283–288.

[12] S. Ajmera and T. Pattanshetti, "A Survey Report on Identifying Different Machine Learning Algorithms in Detecting Domain Generation Algorithms within Enterprise Network," in *Proceedings of 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–5.

[13] H. Suryotrisongko, Y. Musashi, A. Tsuneda, and K. Sugitani, "Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing," *IEEE Access*, vol. 10, pp. 34 613–34 624, 2022.

[14] M. A. Ayub, S. Smith, A. Siraj, and P. Tinker, "Domain Generating Algorithm based Malicious Domains Detection," in *Proceedings of the 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, June 2021, pp. 77–82.

[15] Y. Zhang, Y. Wu, and S. Jin, "Which DGA Family does A Malicious Domain Name Belong To," in *Proceedings of the 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, July 2020, pp. 53–60.

[16] R. R. Curtin, A. B. Gardner, S. Grzonkowski, A. Kleymenov, and A. Mosquera, "Detecting DGA Domains with Recurrent Neural Networks and Side Information," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES '19, 2019.

[17] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A Comprehensive Measurement Study of Domain Generating Malware," in *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 263–278.

[18] C. Tianqi and G. Carlos, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, p. 785–794.

[19] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 30, 2017.

[20] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 31, 2018.

[21] R. Johnson and T. Zhang, "Learning Nonlinear Functions Using Regularized Greedy Forest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 942–954, 2014.

[22] "The DGA of PadCrypt," https://bin.re/blog/the-dga-of-padcrypt/.

[23] "The DGA of Sisron," https://bin.re/blog/the-dga-of-sisron/.

[24] "Tinba's DGA Adds Other Top Level Domains," https://bin.re/blog/new-top-level-domains-for-tinbas-dga/.

[25] A. Mills and P. Legg, "Investigating Anti-Evasion Malware Triggers Using Automated Sandbox Reconfiguration Techniques," *Journal of Cybersecurity and Privacy*, vol. 1, no. 1, pp. 19–39, 2021.

[26] "The DGA of Zloader," https://bin.re/blog/the-dga-of-zloader/.

[27] L. Csikor, H. Singh, M. S. Kang, and D. M. Divakaran, "Privacy of DNS-over-HTTPS: Requiem for a Dream?" in *Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021, pp. 252–271.

[28] D. Vekshin, K. Hynek, and T. Cejka, "DoH Insight: Detecting DNS over HTTPS by Machine Learning," in *Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES)*, 2020.

[29] "Cloudflare Tunnel (Cloudflared)," https://developers.cloudflare.com/cloudflare-one/connections/connect-apps.

[30] "Amazon Alexa Voice AI," https://developer.amazon.com/en-US/alexa/.

[31] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, "Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic," in *Proceedings of 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2020, pp. 63–70.

[32] "CIRA-CIC-DoHBrw-2020," https://www.unb.ca/cic/datasets/dohbrw-2020.html.

[33] R. Mitsuhashi, A. Satoh, Y. Jin, K. Iida, T. Shinagawa, and Y. Takai, "Identifying Malicious DNS Tunnel Tools from DoH Traffic Using Hierarchical Machine Learning Classification," in *Proceedings of 24th International Conference on Information Security (ISC 2021)*, 2021, pp. 238–256.

[34] Y. Khodjaeva and N. Zincir-Heywood, "Network flow entropy for identifying malicious behaviours in dns tunnels," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, ser. ARES 2021, 2021.

[35] "Dohlyzer," https://github.com/ahlashkari/DoHlyzer.

[36] S. R, S. S. Ayachit, V. Patil, and A. Singh, "Competitive Analysis of the Top Gradient Boosting Machine Learning Algorithms," in *Proceedings of 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 191–196.

[37] "Regularized Greedy Forest," https://github.com/RGF-team/rgf.

[38] "VirusShare," https://virusshare.com/.

[39] "The DoH-DGA-Malware-Traffic-HKD dataset," https://github.com/rikima-mitsuhashi/DoH-DGA-Malware-Traffic-HKD.