DomainProfiler: toward accurate and early discovery of domain names abused in future

Daiki Chiba¹ · Takeshi Yagi¹ · Mitsuaki Akiyama¹ · Toshiki Shibahara¹ · Tatsuya Mori² · Shigeki Goto²

Published online: 16 December 2017

© The Author(s) 2017. This article is an open access publication

Abstract

Domain names are at the base of today's cyber-attacks. Attackers abuse the domain name system (DNS) to mystify their attack ecosystems; they systematically generate a huge volume of distinct domain names to make it infeasible for blacklisting approaches to keep up with newly generated malicious domain names. To solve this problem, we propose DOMAINPROFILER for discovering malicious domain names that are likely to be abused in future. The key idea with our system is to exploit *temporal variation patterns (TVPs)* of domain names. The TVPs of domain names include information about how and when a domain name has been listed in legitimate/popular and/or malicious domain name lists. On the basis of this idea, our system actively collects historical DNS logs, analyzes their TVPs, and predicts whether a given domain name will be used for malicious purposes. Our evaluation revealed that DOMAINPROFILER can predict malicious domain names 220 days beforehand with a true positive rate of 0.985. Moreover, we verified the effectiveness of our system in terms of the benefits from our TVPs and defense against cyber-attacks.

Keywords Network-level security and protection · Domain name · DNS · Malware · Temporal variation pattern

1 Introduction

Domain names are used by all Internet users and service providers for their online activities and businesses. Domain names and their protocol [domain name system (DNS)] are

one of the most successful examples of distributed systems that can satisfy users' needs regarding easy use of the Internet. However, Internet users also include attackers who abuse easy-to-use domain names as a reliable cyber-attack infrastructure. For example, in today's cyber-attacks, domain names are used in serving malicious content or malware, controlling malware-infected hosts, and stealing personal or important information.

As countermeasures against domain name abuses, detecting and blacklisting known malicious domain names are basic strategies that are widely applied to protect users from cyber-attacks. However, attackers understand these countermeasures and abuse DNS to mystify their attack ecosystems; DNS fast-flux and domain generation algorithms (DGAs) are used to evade blacklisting. The key feature of these techniques is that they systematically generate a huge volume of distinct domain names. These techniques have made it infeasible for blacklisting approaches to keep up with newly generated malicious domain names.

Ideally, to fully address the underlying problem with domain name blacklists, we need to observe and track all newly registered and updated domain names in real time and judge whether they are involved in any attackers' infrastructure. However, in reality, this problem is virtually impossible

This paper is the extended version of the paper presented at IEEE/IFIP DSN 2016 [15].

✉ Daiki Chiba
daiki.chiba@ieee.org

Takeshi Yagi
yagi.takeshi@lab.ntt.co.jp

Mitsuaki Akiyama
akiyama@ieee.org

Toshiki Shibahara
shibahara.toshiki@lab.ntt.co.jp

Tatsuya Mori
mori@nsl.cs.waseda.ac.jp

Shigeki Goto
goto@goto.info.waseda.ac.jp

¹ NTT Secure Platform Laboratories, 3-9-11, Midori-cho, Musashino-shi, Tokyo, Japan

² Waseda University, 3-4-1, Okubo, Shinjuku-ku, Tokyo, Japan

to solve for three reasons. One is that attackers use techniques, such as DNS fast-flux and DGAs, to systematically generate a huge volume of distinct domain names. The second is that the number of existing domain names is too large to track in real time. The number of second-level domain (2LD) names (e.g., `example.com`) is now over 296 million [44]. Multiple fully qualified domain names (FQDNs) (e.g., `www.example.com`) may exist under the same 2LD names; therefore, the number of all existing FQDNs could be in the billions. The third reason is that no one can fully understand all real-time changes in the mappings between domain names and IP addresses. Since DNS is a distributed system and the mappings are configured in each authoritative name server, the mappings of all domain names cannot feasibly be observed in real time. Given these reasons, black-listing approaches based on DNS observations have failed to keep up with newly generated malicious domain names. Thus, we adopt an approach of *prediction* instead of observation, i.e., we aim to discover malicious domain names that are likely to be abused in future. The key idea of this approach is to exploit *temporal variation patterns (TVPs)* of malicious domain names. The TVPs of domain names include the information about how and when a domain name has been listed in legitimate/popular and/or malicious domain name lists. We use TVPs to comprehend the variations in domain names. For example, a domain name may be newly registered or updated, IP addresses corresponding to the domain name may be changed, and the traffic directed to the domain name may be changed.

On the basis of the aforementioned idea, we developed a system that actively collects historical DNS logs, analyzes their TVPs, and predicts whether a given domain name will be used maliciously. Our main contributions are summarized as follows.

- We propose DOMAINPROFILER, which identifies TVPs of domain names to precisely profile various types of malicious domain names.
- Our evaluation with real and large ground truth data reveals that DOMAINPROFILER can predict malicious domain names 220 days beforehand with a true positive rate (TPR) of 0.985 in the best-case scenario.
- We reveal the contribution or importance of each feature in DOMAINPROFILER to detect future malicious domain names.
- We conduct a lifespan analysis for malicious domain names detected by DOMAINPROFILER to illustrate the characteristics of various domain names abused in a series of cyber-attacks.
- We use a large number of actual malware samples to demonstrate the effectiveness of DOMAINPROFILER at defending against malware activities.

The rest of this paper is organized as follows. We give the motivation for our key idea in Sect. 2. In Sect. 3, we discuss our proposed system, DOMAINPROFILER. We describe the datasets we used and the results of our evaluation in Sect. 4. We discuss the limitations of our system in Sect. 5 and related work in Sect. 6. Finally, we conclude our paper in Sect. 7.

2 Motivation: temporal variation pattern

We define a temporal variation pattern (TVP) as the time series behavior of each domain name in various types of domain name lists. Specifically, we identify how and when a domain name has been listed in legitimate/popular and/or malicious domain name lists. Our motivation for considering TVPs is based on the observation that both legitimate and malicious domain names vary dramatically in domain name lists over time. There are three reasons for using different and multiple domain name lists. One is that the data are realistically observable; that is, we can easily access the data from domain name list maintainers. The second is that domain name lists are created on the basis of objective facts confirmed by the maintainer of those lists. The third is that multiple domain name lists and the time series changes in those lists can boost the reliability of listed domain names.

As shown in Fig. 1, our proposed system defines and identifies four TVPs (*null*, *stable*, *fall*, and *rise*) for each domain name in a domain name list. *Null* means the domain name has not been listed in the specified time window. *Stable* means the domain name has been continuously listed in the time window. *Fall* is a state in which the domain name was first listed and then delisted during the time window. *Rise* means that the domain name was first unlisted and then listed during the time window.

Definition A set $T_d = \{t_1, \dots, t_{N_d}\}$ is an ordered N_d of timestamps when a domain name d has been listed/contained in a domain name list. The domain name list is collected from t_s to t_e . Given a set of timestamps T_d and a time window between a starting point w_s and ending point w_e , the TVP of a domain name is defined as follows.

Fig. 1 Simplified temporal variation patterns (TVPs)

$$\text{TVP} = \begin{cases} \text{Null} & (\min(T_d \cup \{t_e\}) > w_e \vee \max(T_d \cup \{t_s\}) < w_s) \\ \text{Stable} & (\min(T_d \cup \{t_e\}) < w_s \wedge \max(T_d \cup \{t_s\}) > w_e) \\ \text{Fall} & (\min(T_d \cup \{t_e\}) < w_s \wedge w_s < \max(T_d \cup \{t_s\}) < w_e) \\ \text{Rise} & (w_s < \min(T_d \cup \{t_e\}) < w_e \wedge \max(T_d \cup \{t_s\}) > w_e) \end{cases}$$

These TVPs are common and generic features that can contribute to accurately discriminating malicious domain names controlled by attackers from legitimate domain names. Thus, the focus of these patterns covers a wide range of malicious domain names used in a series of cyber-attacks such as drive-by download attacks, malware downloads, command and control (C&C), and phishing attacks.

In this paper, we use the domain names ranked in the Alexa Top Sites [1] as the legitimate/popular domain name list. Alexa lists the top one million most popular sites on the basis of their global one-month average traffic ranks. We divide the Alexa list on the basis of the ranks to create four domain name lists, Alexa top 1000 (*Alexa1k*), Alexa top 10,000 (*Alexa10k*), Alexa top 100,000 (*Alexa100k*), and Alexa top 1,000,000 (*Alexa1M*). The TVPs for the Alexa Top Sites are identified on the basis of these four lists. Figure 2 shows examples of typical domain names that fit the four patterns in Alexa1M. The graph indicates the relationships between domain names and their Alexa rank variations over time (note the logarithmic y-axis). In the null pattern of Alexa1M (*Alexa1M-Null*), the rank of a domain name has always been outside 1M and has never been listed in Alexa1M. The Alexa1M-Null pattern is intended to be one of the features or hints to boost true positive rates (TPRs), which is the ratio of correctly predicted malicious domain names to actual malicious domain names. This is because the rank for legitimate domain names is more likely to be within 1M, and new domain names by attackers cannot be in Alexa1M right after they have been registered. In the stable pattern of Alexa1M (*Alexa1M-Stable*), the rank of a domain name has always been within 1M and listed in Alexa1M. Alexa1M-Stable includes stable popular domain names; thus, this pattern can be used for improving true negative rates (TNRs), which is the ratio

of correctly predicted legitimate domain names to actual legitimate domain names. In the fall pattern of Alexa1M (*Alexa1M-Fall*), the rank of a domain name was first within 1M, fell, and finally was delisted from Alexa1M. To improve TPRs, the Alexa1M-Fall pattern is intended to detect maliciously re-registered, parked, and hijacked domain names that changed from originally legitimate domain names. In the rise pattern of Alexa1M (*Alexa1M-Rise*), the rank of a domain name was first outside 1M and then increased to be within 1M. This Alexa1M-Rise pattern includes legitimate start-up Web sites' domain names during the specified time window to improve TNRs.

We use the domain names listed in the public blacklist hpHosts [2] as the malicious domain name list. hpHosts provides malicious domain names of malicious Web sites engaged in exploits, malware distribution, and phishing. The TVPs for hpHosts are defined similarly for Alexa. Note that hpHosts does not have any continuous value, such as ranking, and only has information of whether domain names are listed. Figure 3 shows examples of typical domain names that fit the four patterns in hpHosts. In the null pattern of hpHosts (*hpHosts-Null*), a domain name has never been listed in hpHosts. This hpHosts-Null pattern can be used for improving TNRs because legitimate domain names are less likely to be listed in hpHosts. In the stable pattern of hpHosts (*hpHosts-Stable*), a domain name has always been listed in hpHosts. To improve TPRs, the hpHosts-Stable pattern includes domain names related to bullet-proof hosting providers, which provide network resources even to attackers. In the fall pattern of hpHosts (*hpHosts-Fall*), a domain name was once listed and then delisted. For example, this pattern includes domain names that were once abused and then

Fig. 2 Popular domain name list (Alexa top sites)

Fig. 3 Example of TVPs in malicious domain name list (hpHosts)

Table 1 Relationships between TVPs and objectives

TVPs	Objectives
Alexa-null	Improving true positive rates (TPRs)
Alexa-stable	Improving true negative rates (TNRs)
Alexa-fall	Improving true positive rates (TPRs)
Alexa-rise	Improving true negative rates (TNRs)
hpHosts-null	Improving true negative rates (TNRs)
hpHosts-stable	Improving true positive rates (TPRs)
hpHosts-fall	Improving true negative rates (TNRs)
hpHosts-rise	Improving true positive rates (TPRs)

sanitized to improve TNRs. In the rise pattern of hpHosts (*hpHosts-Rise*), a domain name was listed from the middle of the specified time window. This hpHosts-Rise pattern is intended to detect newly registered malicious domain names that attackers will use for a while. Specifically, many sub-domain names can be created under the same domain name to bypass fully qualified domain names (FQDN)-level black-lists. Thus, the hpHosts-Rise pattern contributes to clarifying the situation to increase TPRs.

As described above, these TVPs in both legitimate or popular and malicious domain name lists contribute to boosting both TPRs and TNRs. Table 1 summarizes the relationships between the TVPs and their objectives. The effectiveness of using these patterns in real datasets is described later in Sect. 4.

3 Our system: DomainProfiler

DOMAINPROFILER identifies the temporal variation patterns (TVPs) of domain names and detects/predicts malicious domain names. Figure 4 gives an overview of our system architecture. DOMAINPROFILER is composed of two major modules: monitoring and profiling. The monitoring module collects various types of essential data to evaluate the maliciousness of unknown domain names. The profiling module detects/predicts malicious domain names from inputted target domain names by using the data collected with the monitoring module. The details of each module are explained step-by-step in the following subsections.

3.1 Monitoring module

The monitoring module collects three types of information that will be used later in the profiling module. The first is domain name lists. As discussed in Sect. 2, we need to collect the legitimate/popular domain name list (Alexa) and malicious domain name list (hpHosts) daily to create

Fig. 4 Overview of our system

a database of listed domain names and their time series variations.

The second is historical DNS logs, which contains time series collections of the mappings between domain names and IP addresses. A passive DNS [46] is one typical way to collect such mappings by storing resolved DNS answers at large caching name servers. Due to the privacy policy of our organization, we do not use the passive DNS approach. Instead, we actively send DNS queries to domain names to monitor and build a passive DNS-like database. On the plus side, this active monitoring contains no personally identifiable information of the senders. Moreover, we can control DNS queries so as not to contain disposable domain names [13], which are non-informative and negatively affect the database. For example, disposable domain names are one-time domain names automatically generated to obtain a user's environmental information by using certain antivirus products and web services. Since these domain names are distinct, the mappings between domain names and IP addresses significantly increase the database size with non-informative information for evaluating the maliciousness of domain names. On the minus side of active monitoring, we can only query known domain names and cannot gather the mappings of unknown domain names. Thus, to partially address this problem, we have expanded known existing domain names as much as possible. For example, we have extracted all domain names in domain name lists such as Alexa and hpHosts. Moreover, we crawl approximately 200,000 web pages every day to gather web content and extract domain names. Furthermore, we query a search engine API (2.5M queries/month) to expand the domain names on the basis of the above results.

The third type of information is the ground truth, which will be used to label the training dataset and evaluate the effectiveness of our system. Our ground truth includes the results of web client-based honeypots (honeyclients) and sandbox systems and some subscription-based data such as VirusTotal [3] and professional services by a security vendor. The details of the ground truth we used are given later in Sect. 4.1.

Fig. 5 Definition of domain name terms

3.2 Profiling module

The profiling module consists of three steps that use the information collected from the monitoring module to finally output malicious domain names from inputted target domain names.

3.2.1 Step 1: identifying TVPs

Step 1 identifies the TVPs for each input target domain name. (The definition of a TVP was given in Sect. 2.) First, we query the input domain name to the domain name lists database to obtain the time series data of listed domain names that match the second-level domain (2LD) part of the input domain name. The database consists of five domain name lists: Alexa1k, Alexa10k, Alexa100k, Alexa1M, and hpHosts.

To precisely define the TVP of every domain name, we define that the top-level domain (TLD) includes an effective TLD or public suffix [34] such as .com.au, .co.jp, and .co.uk, as shown in Fig. 5. In general, TLDs are divided into generic top-level domains (gTLDs), such as .com, .net, and .org, and country code top-level domains (ccTLDs), such as .au, .jp, and .uk. If we do not use effective TLDs, the 2LD parts of gTLDs and ccTLDs differ significantly. For example, in the gTLD case of foo.bar.example.com, the 2LD part is example.com; however, in the ccTLD case of baz.qux.example.co.jp, the 2LD part is co.jp. Our definition of including effective TLDs is intended to treat both gTLD and ccTLD identically, that is, the 2LD part in the above ccTLD example is example.co.jp in this paper.

Second, the TVPs of the matched 2LD parts within a specified time window are identified using the predefined patterns (null, stable, fall, and rise), as shown in Sect. 2.

Third, the numbers of matched 2LD parts for the four patterns are counted and used as feature vectors in a machine learning algorithm. Specifically, the feature vectors created in step 1 correspond to Nos. 1–20 of the features listed in Table 2, that is, Nos. 1–4 are for Alexa1k, Nos. 4–8 are for Alexa10k, Nos. 9–12 are for Alexa100k, Nos. 13–16 are for Alexa1M, and Nos. 17–20 are for hpHosts.

3.2.2 Step 2: appending DNS-based features

Step 2 appends DNS-based features to the output of step 1, which are input target domain names with identified TVPs. This step is intended to detect malicious domain names that share common features in terms of IP addresses and domain names. We reviewed and analyzed the typical features proposed for known approaches to select the DNS-based features. The known approaches related to ours are summarized later in Sect. 6. As a result of verifying the availability and effectiveness of the features, we decided to use the features proposed for Notos [5]. The DNS-based features are mainly divided into two types: related IP addresses (rIPs) and related domain names (rDomains).

To acquire features of rIPs, we need to first construct a graph of rIPs for each target domain name. Figure 6 shows an example of rIPs for foo.example.com. The graph is a union of every resolved IP address corresponding to each domain name at the FQDN level and its parent domain name levels, such as 3LD and 2LD, from historical DNS logs collected in the former monitoring module. In Fig. 6, FQDN and 3LD (foo.example.com) correspond to the IP address 192.0.2.2 at time $t - 1$ and 198.51.100.2 at t , and 2LD (example.com) corresponds to the IP address 192.0.2.1 at $t - 1$ and 198.51.100.1 at t . Thus, these four IP addresses are defined as rIPs for foo.example.com. Then, we extract the features from rIPs. These features consist of three subsets: border gateway protocol (BGP), autonomous system number (ASN), and registration.

The BGP features, Nos. 21–29 in Table 2, are created from the information of BGP prefixes corresponding to the related IP addresses (rIPs) of each target domain name. To obtain the required BGP information, we refer to the CAIDA dataset [12]. Specifically, we extract the number of rIPs' BGP prefixes of the target FQDN (No. 21), that of the 3LD part of the target (No. 22), and that of the 2LD part of the target (No. 23); the number of countries for the BGP prefixes of the target FQDN (No. 24), that of the 3LD part of the target (No. 25), and that of the 2LD part of the target (No. 26); the number of rIPs for the 3LD part of the target (No. 27) and that for the 2LD part of the target (No. 28); and the number of organizations for the BGP prefixes of the target FQDN (No. 29).

The ASN features, Nos. 30–32 in Table 2, are created from the ASN information corresponding to the rIPs of each target domain name. To obtain the ASN information, we refer to the MaxMind GeoIP2 databases [32]. Specifically, we extract rIPs' ASNs of the target FQDN (No. 30), that of the 3LD part of the target (No. 31), and that of the 2LD part of the target (No. 32).

The registration features, Nos. 33–38 in Table 2, are created from the IP address registration information corre-

Fig. 6 Graph for related IP addresses (rIPs)

Fig. 7 Graph for related domain names (rDomains)

sponding to the rIPs of each target domain name. To obtain the registration information, we refer to the information of delegated IP addresses [30] from all regional Internet registries (RIRs), namely AFRINIC, APNIC, ARIN, LACNIC, and RIPE NCC. Specifically, we extract the number of RIRs of the rIPs for the target FQDN (No. 33), that of the 3LD part of the target (No. 34), and that of the 2LD part of the target (No. 35) and the diversity or number of allocated dates of the rIPs for the target FQDN (No. 36), that of the 3LD part of the target (No. 37), and that of the 2LD part of the target (No. 38).

On the other hand, to acquire the features of related domain names (rDomains), we need to construct a graph of rDomains for each target domain name using the historical DNS logs collected in the monitoring module. Figure 7 shows an example of rDomains for `foo.example.com`. The graph is a union of domain names pointing to IP addresses in the same autonomous system number (ASN) of the historical IP addresses of each target domain name. In Fig. 7, the ASN for the target `foo.example.com` is AS64501 and another IP address 192.0.2.3 in AS64501 is connected to the domain names `bar.example.net` and `baz.example.org`. Thus, these three domain names are defined as rDomains for `foo.example.com`, and we extract their features. These features consist of three subsets: FQDN string, n-grams, and top-level domain (TLD).

The FQDN string features, Nos. 39–41 in Table 2, are created from the set of rDomains for each target domain name. Specifically, we extract the number of FQDNs (No. 39) in the rDomains, mean length of the FQDNs (No. 40), and standard deviation (SD) of the length of the FQDNs (No. 41).

The n-gram features, Nos. 42–50 in Table 2, are created from the occurrence frequency of n-grams ($n = 1, 2, 3$) in the set of rDomains for each target domain name. Note that

Fig. 8 Random Forests

the units of n-grams in this paper are denoted with letters; thus, 2-grams for `example.com` consists of pairs of letters such as `ex`, `xa`, and `am`. Specifically, we extract the mean, median, and SD of 1-gram (Nos. 42–44) in rDomains, those of 2-grams (Nos. 45–47), and those of 3-grams (Nos. 48–50).

The TLD features, Nos. 51–55 in Table 2, are created from TLDs in the set of rDomains for each target domain name. Specifically, we extract the distinct number of TLDs in the set of rDomains (No. 51), ratio of the `.com` TLD in the set (No. 52), and mean, median, and SD of the occurrence frequency of the TLDs in the set (Nos. 53–55).

3.2.3 Step 3: applying machine learning

Step 3 involves applying a machine learning algorithm to the outputs of step 2, which consist of input target domain names with all the features listed in Table 2. This step is designed to achieve our goal of detecting/predicting domain names that will possibly be used maliciously in future. To this end, we use supervised machine learning to effectively find possible malicious domain names from unvalued input domain names. Supervised machine learning basically consists of two phases: training and test. The training phase generates a learning model on the basis of the labeled malicious and legitimate training data by using extracted features. The test phase uses this learning model to calculate the maliciousness of each input domain name by using the extracted features to detect/predict malicious domain names.

Among many supervised machine learning algorithms, we selected *Random Forests* [10] because of their high accuracy, as identified in our preliminary experiments, and high scalability, which we can easily parallelize. The concept image of Random Forests is shown in Fig. 8. Random Forests consist of many *decision trees*, which are constructed from input data with randomly sampled features. The final prediction is output by the majority vote of the decision trees.

Table 3 Dataset

Type	Dataset	Period	# FQDNs
Target domain names (training set)	Legitimate-alexa	2013-05-22–2015-02-28	89,739
	Malicious-hpHosts	2013-01-17–2015-02-28	83,670
Target domain names (test set)	Honeyclient-exploit	2015-03-01–2015-10-07	537
	Honeyclient-malware	2015-03-01–2015-10-07	68
	Sandbox-malware	2015-03-01–2015-10-07	775
	Sandbox-C&C	2015-03-01–2015-10-07	8473
	Pro-C&C	2015-03-01–2015-03-29	97
	Pro-phishing	2015-03-01–2015-03-29	78,221
	Legitimate-new	2015-03-01–2015-03-29	5868
Domain name lists DB	AlexaDB	2013-05-22–2015-02-28	5,596,219
	hpHostsDB	2013-01-17–2015-02-28	1,709,836
Historical DNS logs	DNSDB	2014-10-01–2015-02-28	47,538,966

4 Evaluation

DOMAINPROFILER was evaluated using real datasets including an extensive number of domain names. This section explains how we evaluated it in terms of its effectiveness at using temporal variation patterns (TVPs) and detecting/predicting malicious domain names used in real cyber-attacks.

4.1 Dataset

Our evaluations required three types of datasets, as shown in Table 3: target domain names, domain name list databases, and historical DNS logs.

The first dataset was target domain names, which were composed of training and test sets. The training set was labeled data for creating a learning model in Random Forests. To create the *Legitimate-Alexa*, we extracted fully qualified domain names (FQDNs) on the basis of the domain names listed in Alexa100k. Since most domain names in Alexa are second-level domain (2LD) names and do not have IP addresses, we used a search engine API to randomly extract existing FQDNs in the wild from each 2LD name. Moreover, as shown in Sect. 3.1, we used our ground truth, such as the results of honeyclients and subscription-based professional data, to eliminate the possibility that malicious domain names are in Legitimate-Alexa. As for *Malicious-hpHosts*, we used a process similar to that for Legitimate-Alexa; that is, we extracted FQDNs from 2LD names listed in hpHosts using a search engine and verified the maliciousness by using our ground truth.

The test set was used for evaluating the predictive detection performance of our system. Note that there were no overlaps between the training and test sets and the collected period of the test set was after that of the training

set. Thus, we can use the test set to simulate the performance of DOMAINPROFILER at predicting domain names that will be abused in future. For web client-based honeypot (honeyclient) datasets, we used our honeyclients to collect newly malicious FQDNs, particularly related to drive-by download attacks, from March to October 2015. In a typical drive-by download attack, a legitimate Web site is compromised to lead users to malicious domain names. In our evaluations, we used two types of malicious domain names owned/managed by attackers. *Honeyclient-Exploit* contained FQDNs of Web sites engaged in distributing exploits that target users' browsers and their plug-ins. *Honeyclient-Malware* was the collection of FQDNs used for malware distribution Web sites in drive-by download attacks. To create sandbox datasets, we used our sandbox systems to run 13,992 malware samples randomly downloaded from VirusTotal [3]. The *Sandbox-Malware* dataset contained FQDNs connected by malware samples (e.g., downloader) to download other malware samples. The *Sandbox-C&C* dataset was a collection of FQDNs of command and control (C&C) servers detected by using our sandbox. The *Pro-C&C* and *Pro-Phishing* datasets were FQDNs used for C&C servers and phishing Web sites. Note that the Pro datasets were obtained from commercial and professional services provided by a security vendor, and the FQDNs we selected were only those that had a high likelihood of being abused by attackers. Furthermore, we prepared entirely different legitimate domain names from other datasets as the *Legitimate-New* dataset. This dataset was used to fairly evaluate false positives when operating DOMAINPROFILER in March 2015. The *Legitimate-New* dataset only contained legitimate domain names observed in a large campus network. We manually checked the domain names and excluded any contained in the other training and test sets.

The second dataset was a domain name list database used for identifying TVPs. As explained in Sect. 2, we selected

Alexa top sites as a legitimate/popular list and hpHosts as a malicious list since they are continuously obtainable daily.

The third dataset was historical DNS logs, which involved time series collections of the domain name and IP address mappings. As discussed in Sect. 3.1, we actively sent DNS queries to the domain names we found by using domain name lists, our web crawler, and search engine API. That is, we extracted all domain names in domain name lists such as Alexa and hpHosts. Moreover, we crawled approximately 200,000 web pages every day to gather web content and extracted domain names from the content and their static and dynamic hyperlinks. Furthermore, we expanded the number of domain names by using an external search engine API (2.5M queries/month) on the basis of the above domain names. In our evaluations, we used over 47M distinct FQDNs and their time series changes from October 2014 to February 2015, as shown in Table 3.

4.2 Parameter tuning

Before we evaluated our DOMAINPROFILER, we needed to tune two types of parameters: the size of the time window in TVPs (step 1) and the required parameters to run the Random Forests (step 3).

Here is the summary of evaluation criteria discussed in the following sections. A true positive (TP) is the number of correctly predicted malicious domain names, a false positive (FP) is the number of incorrectly predicted legitimate ones, a false negative (FN) is the number of incorrectly predicted malicious ones, and a true negative (TN) is the number of correctly predicted legitimate ones. The true positive rate (TPR), otherwise known as recall, is the ratio of correctly detected malicious domain names to actual malicious domain names. The true negative rate (TNR) is the ratio of correctly determined legitimate domain names among actual legitimate domain names. The false positive rate (FPR) is the ratio of incorrectly determined legitimate domain names among actual legitimate domain names. The precision is the ratio of actual malicious domain names to domain names detected as malicious by DOMAINPROFILER. The F-measure is the ratio that combines recall and precision, i.e., it is calculated as the harmonic mean of precision and recall.

4.2.1 Time window size

We conducted tenfold cross-validations (CVs) using the training set with variable time window sizes (from 1 to 365 days) to select the time window size on the basis of the evaluation criteria. Figure 9 shows two graphs: the left one corresponds to the time window sizes from 1 to 365 days and the right one to those from 1 to 7 days. These two graphs reveal that the best time window size for TVPs is only 2 days. This is not a surprising result for us in terms of the nature

Fig. 9 Tuning time window size

Fig. 10 Tuning Random Forests' parameters

of domain names or TVPs because attackers abuse the DNS to generate a huge volume of distinct domain names from one day to the next, so keeping old information over a long period decreases the F-measure of our system.

4.2.2 Random Forests

Random Forests [10] require two parameters to run. One is the number of decision trees. As explained in Sect. 3.2.3, Random Forests consist of multiple decision trees; thus, we needed to decide how many trees to make beforehand. As is the case with the aforementioned time window size, we conducted tenfold CVs by changing the number of trees to determine the optimum number of decision trees. The left graph in Fig. 10 shows the relationships between the number of trees and the F-measure. The graph shows that F-measures are stable over 100 trees. Thus, we decided to use 100 decision trees in the following evaluations.

The other parameter is the number of sampled features in each individual decision tree. Random Forests construct decision trees from input data that have randomly sampled features to improve overall accuracy. We conducted tenfold CVs again to search for the optimum number of sampled features. The right graph in Fig. 10 shows that the best F-measure is obtained when there are seven sampled features.

4.3 Feature set selection

Now that we have selected the optimal parameters (the time window size, number of trees, and number of sampled features in each tree), this section compares the detection performance with different feature sets. The feature sets include

Table 4 Detection performance with different feature sets (cross-validation)

Feature set	TP	FP	FN	TN	# FQDNs	TPR/recall	TNR	FPR	Precision	F-measure
TVP	81,436	834	2234	88,905	173,409	0.973	0.991	0.009	0.990	0.982
rIP	62,734	32,347	20,270	57,306	172,657	0.756	0.639	0.361	0.660	0.705
rDomain	58,095	16,523	24,873	72,893	172,384	0.700	0.815	0.185	0.779	0.737
rIP+rDomain	61,928	13,197	21,033	76,206	172,364	0.746	0.852	0.148	0.824	0.783
TVP+rIP+rDomain	80,879	798	2082	88,605	172,364	0.975	0.991	0.009	0.990	0.983

Fig. 11 ROC curves

the temporal variation pattern (TVP), related IP address (rIP), related domain name (rDomain), combination of rIP and rDomain (rIP+rDomain), and combination of TVP, rIP, and rDomain (TVP+rIP+rDomain). We conducted tenfold CVs using the training set and the optimal parameters by changing the feature sets to estimate how accurately each feature set will perform in theory. Table 4 illustrates the detection performance using the above evaluation criteria. Note that the number of FQDNs varies with feature sets due to the availability of each feature. For example, some domain names have no rIPs and/or rDomains. Also, Fig. 11 shows the receiver operator characteristic (ROC) curves. An ROC curve shows a pair of a FPR and a TPR corresponding to a particular decision cutoff point. Thus, if the ROC curve of a feature set rises more rapidly, this means that the performance of the feature set is better. Table 4 and Fig. 11 show that using our TVP features significantly contributes to achieving better detection performance than using only DNS-based features. Using only DNS-based features (rIP, rDomain, and rIP+rDomain) does not go beyond 0.90 in any evaluation criteria. These results show that using only conventional DNS-based features [5] is insufficient for detecting malicious domain names in current attack ecosystems. However, combining the DNS-based features with our TVP features (TVP+rIP+rDomain) achieves the best results, specifically, a TPR/recall of 0.975, TNR of 0.991, FPR of 0.009, precision of 0.990, and F-measure of 0.983. These results indicate that our key idea based on using TVPs is effective for improving both TPR and TNR exactly as intended.

4.4 System performance

We evaluated the system performance of a prototype version of DOMAINPROFILER. Specifically, we calculated the execution time and data size in each step when we conducted a tenfold CV using the training set with the optimal parameters and best feature set (TVP+rIP+rDomain). Step 1 (identifying TVPs) was executed on a single server with a 10-core 2.2-GHz CPU and 128-GB RAM. The execution time for extracting TVP features from 173,409 FQDNs was 61 s, which was equivalent to 0.0004 s/FQDN. The file sizes of the domain name list database (SQL) were 1.4 GB in Alexa and 300 MB in hpHosts. Step 2 (appending DNS-based features) was executed as a MapReduce job on a Hadoop cluster, which had 2 master servers (16-core 2.4-GHz CPU, 128-GB RAM) and 16 slave servers (16-core 2.4-GHz CPU, 64-GB RAM). The execution time for extracting rIP features from 173,409 FQDNs was 20 h (0.42 s/FQDN) and that for rDomain features was 96 h (1.99 s/FQDN). The file size of the historical DNS logs used for extracting these DNS-based features was 212 GB. Step 3 (applying machine learning) was executed on the same server as step 1. The execution time for one-time training from 156,068 FQDNs was 28 s (0.0001 s/FQDN) and that for the test from 17,341 FQDNs was 8 s (0.0005 s/FQDN). These evaluations prove the basic feasibility of our proposed system and reveal that step 2 requires far more resources and time to execute than steps 1 and 3. The reason for step 2's high cost is the size of the graphs for rIPs and rDomains. Currently, some domain names used by hypergiants, such as Google, Amazon, and Akamai, have a huge number (over 10,000) of rIPs and rDomains. This fact raises the problem of a high cost for extracting conventional DNS-based features [5]. However, from the results explained in Sect. 4.3, this problem will be solved if our system sacrifices 0.002 of its TPR to use the feature set TVP instead of TVP+rIP+rDP. This is a trade-off between system performance and detection performance. Thus, we should configure our system on the basis of this situation.

Table 5 Predictive detection performance of DOMAINPROFILER (feature set: TVP+rIP+rDomain)

Dataset	TP	FP	FN	TN	# FQDNs	TPR/recall	TNR	FPR	Precision	F-measure
Honeyclient-exploit	529	–	8	–	537	0.985	–	–	–	–
Honeyclient-malware	67	–	1	–	68	0.985	–	–	–	–
Sandbox-malware	721	–	54	–	775	0.930	–	–	–	–
Sandbox-C&C	7476	–	997	–	8473	0.882	–	–	–	–
Pro-C&C	92	–	5	–	97	0.948	–	–	–	–
Pro-phishing	75,583	–	2638	–	78,221	0.966	–	–	–	–
Legitimate-new	–	142	–	5726	5868	–	0.976	0.024	–	–
Total	84,468	142	3073	5726	94,309	0.958	0.976	0.024	0.998	0.978

Table 6 Predictive detection performance of conventional DNS-based features (feature set: rIP+rDomain)

Dataset	TP	FP	FN	TN	# FQDNs	TPR/recall	TNR	FPR	Precision	F-measure
Honeyclient-exploit	184	–	353	–	537	0.343	–	–	–	–
Honeyclient-malware	5	–	63	–	68	0.074	–	–	–	–
Sandbox-malware	197	–	578	–	775	0.254	–	–	–	–
Sandbox-C&C	2491	–	5982	–	8473	0.294	–	–	–	–
Pro-C&C	39	–	58	–	97	0.402	–	–	–	–
Pro-phishing	29,427	–	48,794	–	78,221	0.376	–	–	–	–
Legitimate-new	–	1038	–	4830	5868	–	0.823	0.177	–	–
Total	32,343	1038	55,828	4830	94,039	0.367	0.823	0.177	0.969	0.532

4.5 Predictive detection performance

We evaluated the predictive detection performance of DOMAINPROFILER; that is, whether we can discover domain names that may be abused in the *future*. The aforementioned evaluations were based on cross-validations (CVs); however, this section focuses on the evaluation of the detection performance of new malicious domain names that first appeared after March 1, 2015, by using only the information as of February 28, 2015. Specifically, we used the training set shown in Table 3 to create a learning model first and then input the test set in Table 3 to evaluate the predictive detection performance. In this evaluation, we set the optimal parameters discussed in Sect. 4.2. The best feature set (TVP+rIP+rDomain) discussed in Sect. 4.3 was compared with the feature set (rIP+rDomain) that had only conventional DNS-based features [5]. Tables 5 and 6 list the evaluation results of using TVP+rIP+rDomain and rIP+rDomain. Note that the six test datasets (Honeyclient-Exploit, Honeyclient-Malware, Sandbox-Malware, Sandbox-C&C, Pro-C&C, and Pro-Phishing) only consist of malicious domain names; thus, there are no false positives (FPs), true negatives (TNs), or their related evaluation criteria in the tables. On the other hand, the other test dataset (Legitimate-New) only consists of legitimate domain names; thus, there are no true positives (TPs) or false negatives (FNs) as explained in Sect. 4.1.

In terms of the true positive rate (TPR/recall), DOMAINPROFILER using the feature set (TVP+rIP+rDomain) achieved extremely high TPRs in all test sets; our system achieved TPRs of 0.985 in Honeyclient-Exploit and Honeyclient-Malware. Moreover, our system accurately detected/predicted command and control (C&C) domain names in Sandbox-C&C and Pro-C&C, while our training set did not include labeled C&C domain names. This is not a surprising result because our TVP is designed to exploit the common characteristics of attackers' domain names. On the other hand, DOMAINPROFILER using only the conventional features (rIP+rDomain) achieved a TPR of 0.402 at best. Comparing these results illustrates that our TVP features successfully contribute to predicting domain names that will be used maliciously in future.

We also evaluated the true negative rate (TPR) and false positive rate (FPR) using the Legitimate-New dataset since the FPR results during the training period explained in Sect. 4.3 cannot be generalized to testing period or predictive detection performance. Tables 5 and 6 show that DOMAINPROFILER using the feature set (TVP+rIP+rDomain) achieved a TNR of 0.976 and a FPR of 0.024, whereas that using conventional features (rIP+rDomain) achieved a TNR of 0.823 and a FPR of 0.177. In total, DOMAINPROFILER achieved an F-measure of 0.978 in the predictive or future performance evaluation. These results indicate that our TVP features contribute to boosting both TPR and TNR.

Table 7 Early detection performance of DOMAINPROFILER (feature set: TVP+rIP+rDomain)

Dataset	days_Min	days_1stQu	days_2ndQu	days_Mean	days_3rdQu	days_Max
Honeyclient-exploit	16	140	176	164.5	197	220
Honeyclient-malware	1	125	205	159.4	212	220
Sandbox-malware	1	108	133	128.2	151	221
Sandbox-C&C	1	38	99	98.31	140	221
Pro-C&C	1	9	15	14.95	21	29
Pro-phishing	1	9	14	14.04	19	28

Table 8 Early detection performance of conventional DNS-based features (feature set: rIP+rDomain)

Dataset	days_Min	days_1stQu	days_2ndQu	days_Mean	days_3rdQu	days_Max
Honeyclient-exploit	19	158	187	172.3	210	219
Honeyclient-malware	1	60	63	108.4	205	213
Sandbox-malware	3	112	133	128.3	153	221
Sandbox-C&C	1	47	108	103.9	144	221
Pro-C&C	1	9	12	14.18	20	29
Pro-phishing	1	11	17	15.21	20	28

We explain in detail how each TVP feature contributes to successful predictions later in Sect. 4.7.

In terms of early detection of future malicious domain names, we investigated when the system can detect such domain names. Specifically, we analyzed the number of days that elapsed from February 28, 2015, when the learning model was created, for malicious domain names to be detected by the system. For example, if the system correctly detected and identified a new malicious domain name on March 7, 2015, the elapsed number of days for the domain name is seven. Tables 7 and 8 show the descriptive statistics of the elapsed days for malicious domain names for each feature set. Note that we only count domain names in the TP of each dataset shown in Tables 5 and 6. The descriptive statistics include the minimum (days_Min); the first quartile (days_1stQu), which means the value cutoff at the first 25% of the data; the second quartile, which is also called the median and is the value cutoff at 50% of the data; the mean (days_Mean); the third quartile (days_3rdQu), which is the value cutoff at 75% of the data; and the maximum (days_Max). Table 7 shows that our proposed system (TVP+rIP+rDomain) can precisely predict future malicious domain names 220 days before the ground truth, such as honeyclients and sandbox systems, and identify them as malicious in the best case. Comparing the above results with Table 8 reveals that the conventional DNS-based feature set (rIP+rDomain) [5] also detects malicious domain names early; however, the number of detected domain names (TP) is quite small as shown in Table 6. We conclude that our proposed system using TVPs outperforms the system using

only the conventional DNS-based feature set from the perspectives of both accuracy and earliness.

4.6 Effectiveness of each feature

In this section, we analyzed how each feature shown in Table 2 contributes to accurate detection of malicious domain names. To this end, we calculated the importance of each feature in the TVP, rIP, and rDomain in our trained model. Specifically, we used the criterion called the Gini Index (GI) in Random Forests [10]. Random Forests have a mechanism to evaluate the importance of each feature by adding up the GI or *impurity* decreases for each feature over all decision trees constructed in the algorithm of Random Forests. The GI calculation is explained in more detail elsewhere [11,27,28].

Table 9 illustrates our analysis results including the GI score and its rank for each feature over all 55 features listed in Table 2. A higher GI score means that the corresponding feature is more important when detecting malicious domain names. The rank provides a ranking of each feature's contribution among all features. Overall, our TVP features (Nos. 1–20) had more contributions and higher GI scores than both rIP features (Nos. 21–38) and rDomain features (Nos. 39–55). Specifically, 9 features in TVP ranked between 1 and 10 (top 10): Alexa100k-null (No. 9), Alexa100k-stable (No. 10), Alexa100k-fall (No. 11), Alexa1m-null (No. 13), Alexa1m-stable (No. 14), Alexa1m-fall (No. 15), hpHosts-null (No. 17), hpHosts-stable (No. 18), and hpHosts-fall (No. 19). The results for the greater contributions of TVP features are also supported by the previous evaluation results shown in Sect. 4.3. We further detail the case studies of how

Table 9 Contribution of each feature

No.	Feature (TVP)	GI	Rank	No.	Feature (rIP)	GI	Rank	No.	Feature (rDomain)	GI	Rank
1	Alexa1k-null	76.7	44	21	# BGP Prefixes (FQDN)	135.0	40	39	# FQDNs	210.6	21
2	Alexa1k-stable	37.7	51	22	# BGP Prefixes (3LD)	159.5	33	40	Mean lengths	196.1	25
3	Alexa1k-fall	19.4	53	23	# BGP Prefixes (2LD)	261.4	15	41	SD lengths	214.6	20
4	Alexa1k-rise	6.8	55	24	# Countries (FQDN)	156.2	35	42	Mean distribution	163.3	32
5	Alexa10k-null	155.6	36	25	# Countries (3LD)	203.1	23	43	Median distribution	179.5	27
6	Alexa10k-stable	70.3	46	26	# Countries (2LD)	148.2	37	44	SD distribution	174.9	29
7	Alexa10k-fall	48.0	50	27	# IP addresses (3LD)	319.4	13	45	Mean distribution	204.6	22
8	Alexa10k-rise	11.4	54	28	# IP addresses (2LD)	328.3	12	46	Median distribution	189.7	26
9	Alexa100k-null	23197.3	1	29	# Organizations (FQDN)	53.0	48	47	SD distribution	167.9	31
10	Alexa100k-stable	5073.1	5	30	# ASN (FQDN)	129.9	41	48	Mean distribution	177.7	28
11	Alexa100k-fall	2977.5	7	31	# ASN (3LD)	351.8	10	49	Median distribution	85.6	42
12	Alexa100k-rise	330.1	11	32	# ASN (2LD)	215.1	19	50	SD distribution	174.4	30
13	Alexa1m-null	3503.1	6	33	# Registries (FQDN)	35.8	52	51	# TLDs	157.7	34
14	Alexa1m-stable	8255.9	3	34	# Registries (3LD)	49.3	49	52	# Ratio of .com	242.9	17
15	Alexa1m-fall	1068.8	9	35	# Registries (2LD)	74.2	45	53	Mean distribution	252.1	16
16	Alexa1m-rise	60.6	47	36	# Dates (FQDN)	141.5	38	54	Median distribution	140.5	39
17	hpHosts-null	17248.9	2	37	# Dates (3LD)	309.0	14	55	SD distribution	234.6	18
18	hpHosts-stable	5404.1	4	38	# Dates (2LD)	203.0	24				
19	hpHosts-fall	2861.2	8								
20	hpHosts-rise	79.0	43								

each TVP feature contributes to accurate detection later in Sect. 4.7.

Now we focus on features that had lower GI scores and ranked between 46 and 55 (the bottom 10). The underlying nature of the Alexa ranking is the reason for the six low ranked TVP features using Alexa1k and Alexa10k: Alexa1k-stable (No. 2), Alexa1k-fall (No. 3), Alexa1k-rise (No. 4), Alexa10k-stable (No. 6), Alexa10k-fall (No. 7), and Alexa10k-rise (No. 8). Specifically, top 10k in Alexa [1] only contains domain names of extremely popular Web sites, and our datasets shown in Sect. 4.1 did not contain such easy-to-answer domain names. As a result, the importance of these TVP features using Alexa1k and Alexa10k is relatively low. As for Alexa1m-rise (No. 16), the number of domain names matching this TVP is considerable smaller than the total number of domain names in our dataset. In terms of low ranked rIP features such as # Organizations (FQDN) (No. 29), # Registries (FQDN) (No. 33), and # Registries (3LD) (No. 34), the reason is the difference between malicious domain names in the 2000s and those currently used. The conventional rIP features were originally developed and evaluated in 2009 [5]. However, today's malicious domain names abuse cloud hosting services much more than in 2009 as these services have gained in popularity. For example, in a typical cloud hosting service, IP addresses or BGP prefixes are shared among multiple domain names. Thus, the importance of the above three rIP features resulted in lower GI scores. To the best

of our knowledge, we are the first to show these quantitative and detailed results about conventional DNS-based features and their effectiveness for detecting today's malicious domain names. We believe these results will be useful for other researchers to develop new domain reputation systems in future.

4.7 Effectiveness of our temporal variation patterns

We further analyzed how our temporal variation pattern (TVP) features contribute to increasing the true positive rate (TPR) and true negative rate (TNR) simultaneously. We present some noteworthy case studies in our TVPs defined in Sect. 2. In this analysis, we used the TVP+ rIP+rDomain feature set and the same settings and dataset as the previous evaluation discussed in Sect. 4.5.

Alexa1M-Null: This TVP is intended to boost the TPR, as described in Sect. 2. Our analysis revealed that this TVP was especially effective for malicious domain names using a domain generation algorithm (DGA) and abusing new generic top-level domains (gTLDs) such as .xyz and .solutions. This is because these domain names are less likely to be within Alexa1M.

Alexa1M-Stable: This TVP successfully determined the characteristics of somewhat popular domain names to improve the TNR.

Table 10 Dataset for lifespan analysis

Dataset	# of FQDNs in test set	# of Queried FQDNs	# of Answered FQDNs
Honeyclient-exploit	537	537	534
Honeyclient-malware	68	68	67
Sandbox-malware	775	775	610
Sandbox-C&C	8473	8473	6079
Pro-C&C	97	97	91
Pro-phishing	78, 221	50	49
Total	88,171	10,000	7429

AlexaIM-Fall: This TVP is designed to detect the changing of malicious domain names to improve the TPR. We observed two major types of malicious domain names that fit this TVP. One type is expired domain names due to the termination of services or the merger and acquisition of companies. Some of these expired domain names were re-registered by third-party attackers to execute domain parking or cyber-attacks. The other type is domain names that were changed from legitimate to malicious because the Web site of the domain name was not well managed and had poor security.

AlexaIM-Rise: This TVP attempts to determine the features of legitimate domain names of start-up Web sites to improve the TNR. We observed many domain names corresponding to this TVP such as those of new companies, services, movies, and products.

hpHosts-Null: This TVP successfully improved the TNR because the second-level domain (2LD) parts of popular/legitimate domain names were less likely to be listed in hpHosts.

hpHosts-Stable: This TVP is designed to determine the characteristics of malicious domain names abusing easy-to-use services, such as bullet-proof hosting, to improve the TPR. For example, we observed many subdomains using a domain generation algorithm (DGA) under the same 2LD part such as 84c7zq.example.com.

hpHosts-Fall: This TVP is intended to boost the TNR. We confirmed that some domain names under well-managed networks fit this TVP because these domain names were once abused and then quickly sanitized. In this case, the TVP contributed to the accurate prediction of future legitimate domain names.

hpHosts-Rise: This TVP is designed to help detect or predict malicious domain names more accurately to improve the TPR. We mainly observed two types of domain names that fit this TVP. One is domain names that heavily used the DGA in both 2LD and third-level domain (3LD) parts of domain names, e.g., 14c2c5h8[masked].yr7w2[masked].com. We observed that this type of 2LD will be continuously used for a while by attackers to create many subdomain names. The other is domain names under free subdomain

name services, which offer subdomain name creation under 2LD parts, such as .flu.cc and .co.nr. These services are easily abused by attackers for creating distinct domain names.

4.8 Lifespan of detected domain names

We analyzed the lifespan of each malicious domain name detected by DOMAINPROFILER to show the characteristics of domain names abused in various types of cyber-attacks. To this end, we used logs collected from a set of large-scale DNS cache servers. The data are called a passive DNS database (DNSDB) [17], which records various information about a domain name, e.g., a list of resolved IP addresses, history of the accesses to the domain name, and first/last seen timestamp. Due to a limitation of our API usage in the DNSDB, we randomly sampled 10,000 FQDNs from our test set shown in Table 3. The detailed numbers of the selected FQDNs are shown in Table 10. We used all FQDNs of Honeyclient-Exploit, Honeyclient-Malware, Sandbox-Malware, Sandbox-C&C, and Pro-C&C, whereas we randomly selected only 50 FQDNs of Pro-Phishing. We queried the 10,000 FQDNs to the passive DNSDB and obtained the results for 7429 FQDNs. The passive DNSDB has no results for the remaining 2571 FQDNs. One reason for this low cover rate is due to the data collection points in the passive DNSDB. Since it does not cover all of cache servers in the world, some FQDNs created using the DGA or those used only for specific targets (e.g., Advanced Persistent Threat (APT)) are not covered.

Using the results from the passive DNSDB, we conducted lifespan analysis of the malicious domain names. In this evaluation, a lifespan of a domain name is defined to be the period from the first seen timestamp to the last seen timestamp; that is, the period in which the domain name continually has corresponding IP addresses. To precisely analyze the lifespan from the passive DNSDB results, we applied a survival analysis method based on the Kaplan-Meier estimator [23]. The reason for using this method is the statistical characteristics of the lifespan data. Specifi-

Fig. 12 Survival analysis plot for domain names in our test set

cally, the lifespan data are considered to be *right-censored*; that is, the collection period for each domain name is not the same and some domain names remain *alive* at the end of the data collection period. Our survival analysis results for each test set are summarized in Fig. 12. The x-axis shows elapsed time (days) from a first seen timestamp, and the y-axis is the survival probability that a domain name is still alive and has IP addresses after the elapsed time. Figure 12 clearly reveals that the survival probability or the lifespan is very different for each test set; in particular, both Honeyclient-Exploit and Honeyclient-Malware have much shorter lifespans than the others. This is because each test set contains domain names engaged in different types of cyber-attacks such as drive-by download attacks (Honeyclient-Exploit and Honeyclient-Malware), additional malware download (Sandbox-Malware), C&C (Sandbox-C&C, Pro-C&C), and phishing (Pro-Phishing), as stated in Sect. 4.1. These results also indicate that our TVP in DOMAINPROFILER successfully covered a wide range of malicious domain names used in a series of cyber-attacks as intended.

We further analyzed what causes the differences in survival probabilities or lifespans. First, we selected domain names that have shorter lifespans such as Honeyclient-Exploit and Honeyclient-Malware. These domain names were abused in a series of drive-by download attacks. Exploit kits used in drive-by download attacks are designed to avoid their exploits from being analyzed [16]. As a result, attack-

ers need to churn their domain names used for exploit kits. Grier et al. [19] also reported that the lifespan of domain names used for distributing exploit kits is very short. This specific attack characteristic leads to a shorter lifespan in our Honeyclient-Exploit and Honeyclient-Malware dataset. Second, we analyzed domain names that have longer lifespans, namely Sandbox-Malware, Sandbox-C&C, Pro-C&C, and Pro-Phishing. Our analysis revealed that there were some reasons behind the longer lifespan. One reason is *parking* services. A parking domain name is mainly used for displaying advertisements [4,26,45]. Domain names abused in cyber-attacks tend to use such parking services later to monetize the malicious traffic from malware-infected hosts [26]. We detected such parking domain names in our dataset by following the method proposed by Vissers et al. [45]. Our detection results are summarized in Table 11. The results indicate that some domain names, especially in Sandbox-Malware, Sandbox-C&C, Pro-C&C, and Pro-Phishing, were found to currently use such parking services. Please note that we adopted only reliable detection patterns; thus, the number is lower-bound. Another reason behind the longer lifespan is the *sinkhole* operation. A sinkholed domain name is an originally malicious domain name that is treated and then controlled by security organizations [24,36]. In such a case, an IP address is continuously associated with a domain name; thus, the lifespan of the domain name is long. To detect such sinkholed domain names, we followed the same approach originally proposed by Kührer et al. [24]. Table 11 shows the results of our sinkhole detection. We found that the only C&C domain names are operated as sinkhole in our dataset. Please note that we only adopted reliable detection patterns; thus, the number is lower-bound. To explore other reasons, we randomly chose and manually analyzed 100 FQDNs with long lifespans. Seventeen showed some sort of default pages offered by web hosting services, 10 were used in parking services that were not detected by the parking detection patterns described above, 6 were continuously used for C&C servers that offer a certain type of attack command or fingerprint results, and the other 67 had returned connecting errors as of September 2016.

Table 11 FQDNs used for parking and sinkhole services

Dataset	# of Answered FQDNs	# of Parking FQDNs	# of Sinkhole FQDNs
Honeyclient-exploit	534	3	0
Honeyclient-malware	67	2	0
Sandbox-malware	610	82	0
Sandbox-C&C	6079	600	72
Pro-C&C	91	12	6
Pro-phishing	49	12	0
Total	7429	711	78

Table 12 Defended-against malware download activities using DOMAINPROFILER

	Malware family	# Blocked FQDNs	# Blocked samples	First submission date
1	bladabindi	33	2	2016/01/01–2016/01/03
2	gamarue	16	12	2015/10/08–2016/07/29
3	banload	12	23	2015/10/19–2016/08/14
4	ramnit	10	50	2015/10/14–2016/09/15
5	zusy	8	526	2015/10/08–2016/07/16
6	zbot	7	431	2015/10/08–2016/09/11
7	upatre	5	1406	2015/10/08–2016/09/15
8	delf	5	637	2015/10/09–2016/09/01
9	badur	5	6	2015/12/14–2016/08/14
10	ymeta	5	3	2015/10/11–2016/3/5
11	yakes	4	303	2015/10/8–2016/9/12
12	barys	3	7	2015/12/7–2016/4/14
13	bayrob	2	3208	2015/10/12–2016/9/15
14	bublik	2	917	2015/10/8–2016/9/15
15	soxgrave	2	6	2016/5/18–2016/5/24
	Total	119	7537	

4.9 Defending against malware activities

In this evaluation, we provide additional research results on the capability of defending against malware activities using malicious domain names detected using DOMAINPROFILER. Specifically, we analyzed what types of malware download and C&C activities we could defend using DOMAINPROFILER's output. To this end, we created a new sandbox dataset by randomly downloading 354,953 malware samples from VirusTotal [3] and running them in our sandbox system between October 2015 and September 2016. To ensure a fair evaluation, we only used new malware samples that were first seen after October 8, 2015; that is, no data overlapped with those in the datasets shown in Table 3. All malware samples in the dataset were with their malware family information based on 57 different antivirus vendors' scan results. To resolve labeling difficulties such as the lack of a standard naming convention and different family names between antivirus vendors, we used the recently developed tool AVClass [39] to output the most likely malware family name for each malware sample. The total number of malware family names output by AVClass was 310, and we determined that the malware samples and their corresponding malware family names are not biased.

Table 12 lists the top 15 defended-against malware download activities in terms of the number of blocked FQDNs provided by DOMAINPROFILER. In this case, the malware download activities mean that malware samples (e.g., downloader) connect to malicious FQDNs to download other malware samples. Table 12 also shows that the number of malware samples we could block under the condition of blacklisting the malicious FQDNs detected using DOMAIN-

PROFILER as of February 28, 2015, and the period of the first submission dates of corresponding malware samples in VirusTotal. These results indicate that we could defend against at least 7537 new or future malware downloads using only 119 blacklisted FQDNs provided by DOMAINPROFILER.

In contrast, Table 13 lists the top 15 defended-against malware C&C activities. In this case, the malware C&C activities mean that malware samples connect to malicious FQDNs to communicate with C&C servers. Similar to Table 12, Table 13 also shows the number of blocked FQDNs, the number of blocked samples, and the period of first submission date. These results illustrate that we could defend against at least 6933 future malware C&C communications using only 1313 blacklisted FQDNs provided by DOMAINPROFILER.

The above evaluation results indicate that DOMAINPROFILER could defend against various types of malware activities even seven months after blacklisting and prove the effectiveness of DOMAINPROFILER's design. That is, we focus on common and generic features that can contribute to detecting malicious domain names used in a wide variety of cyber-attacks.

5 Discussion

This section discusses possible evasion techniques against DOMAINPROFILER and problems when using the predicted malicious domain names generated from our system as countermeasures to protect users from cyber-attacks.

Table 13 Defended-against malware C&C activities using DOMAINPROFILER

	Malware family	# Blocked FQDNs	# Blocked samples	First submission date
1	bayrob	218	3208	2015/10/12–2016/09/15
2	upatre	129	1406	2015/10/08–2016/09/15
3	banload	127	23	2015/10/19–2016/08/14
4	zbot	110	431	2015/10/08–2016/09/11
5	delf	105	637	2015/10/09–2016/09/15
6	tinba	89	272	2015/10/08–2016/08/23
7	zusy	88	526	2015/10/08–2016/07/16
8	badur	86	6	2015/12/14–2016/08/14
9	salinity	83	124	2015/10/12–2016/08/03
10	gamarue	67	12	2015/10/8–2016/7/29
11	ramnit	56	50	2015/10/14–2016/9/15
12	scar	42	71	2015/11/16–2016/6/10
13	barys	40	7	2015/12/7–2016/4/14
14	bicololo	37	150	2015/10/10–2016/3/1
15	crowti	36	10	2015/11/13–2015/11/19
	Total	1313	6933	

5.1 Evading DomainProfiler

DOMAINPROFILER is designed to exploit the temporal variation patterns (TVPs) of malicious domain names used by attackers. There are three possible techniques to evade our system. One is to avoid using domain names as attack infrastructure. If attackers do not use domain names, we can more easily take countermeasures such as just blocking them using IP addresses. The cost of changing IP addresses is much higher than that of changing domain names due to the limited address space. For instance, the address spaces of IPv4 and IPv6 are limited to 32 and 128 bits, respectively. However, domain names can consist of 255 or fewer octets/characters [33], which means a maximum of a 2040-bit space.

Another evasion technique is to avoid all our features in a TVP, related IP address (rIP), and related domain name (rDomain) to hide malicious domain names from our system. For example, attackers can operate their domain names as real legitimate/popular services for a long time to evade our TVPs and then use the domain names as their malicious infrastructure. However, this situation drives up the cost for implementing any attacks using domain names. Another example is border gateway protocol (BGP) hijacking, which potentially enables attackers to divert user traffic from real IP addresses to their IP addresses. In such a case, attackers may bypass our rIP or rDomain features; however, the BGP is basically used between Internet service providers and is difficult for normal Internet users/attackers to effectively control.

The other possible evasion technique is to use legitimate web services as legitimate users. For example, attackers would create dedicated accounts for some web services and

use them as their command and control (C&C) channels. In such a case, only legitimate domain names are observed and our system cannot detect them. However, these accounts could be easily banned by the administrator of the web services, and the content sent/received by attackers could be easily analyzed to develop a new countermeasure.

5.2 DNS-based blocking

DOMAINPROFILER predicts and outputs malicious domain names. However, these domain names cannot always be blocked with a domain name level. For example, malicious and legitimate Web sites can exist under the same domain name. Thus, blocking on the basis of domain names instead of URLs may excessively block legitimate Web sites. To examine an actual condition, we extracted and checked URLs under each predicted malicious domain name by using a search engine API and commercial ground truth. In this examination, we manually analyzed 250 domain names randomly selected from the Honeyclient-Exploit dataset, as shown in Table 3, by using the following simple heuristics. If multiple URLs are found under the domain name, we consider that the domain name cannot be blocked with a domain name level. On the other hand, if at most one URL is under the domain name, we determine that the domain name can be blocked with a domain name level. The examination results suggest that 72% (=180/250) of domain names can be effectively blocked by using DNS-based blocking without excessive blocking of legitimate Web sites. Therefore, we conclude that malicious domain names output from our system can contribute to expanding DNS-based block lists if we consider the situation of URLs under the domain names.

6 Related work

We summarize known approaches related to ours in terms of evaluating attack infrastructure or resources owned by cyber-attackers. Most of the studies are broadly divided into three approaches: lexical/linguistic, user-centric, and historic relationship. Note that these approaches were often combined in most studies we reviewed; thus, we classify them on the basis of the main idea of each study.

6.1 Lexical/linguistic approach

The lexical/linguistic approach is focused on lexical or linguistic features obtained from malicious attack resources such as URLs and domain names.

Ma et al. [29] proposed a learning approach using features from the lexical structure of malicious phishing URLs. In contrast, our system focuses on not URLs but domain names and can detect not only phishing but also other attacks.

Yadav et al. [47] focused on linguistic features in command and control (C&C) domain names generated using a domain generation algorithm (DGA) and developed an approach for detecting such malicious domain names. Whereas their approach detects C&C domain names containing random strings, our approach targets broader malicious domain names.

Szurdi et al. [41] analyzed the nature of typosquatting domain names. Typosquatting is generally defined as a technique to register similar domain names to popular domain names to profit from advertisements and perform phishing attacks. Although we take a different and more general approach with our system, our temporal variation patterns (TVPs) can also take into account the nature of typosquatting domain names.

Felegyazhi et al. [18] proposed using WHOIS information of domain names such as registration and name servers to detect malicious domain names. Our approach does not use WHOIS information due to the cost of retrieving it; however, we achieve a high true positive rate (TPR) for malicious domain names without WHOIS.

6.2 User-centric approach

The user-centric approach focuses on user behavior of DNS traffic by observing passive DNS logs. Sato et al. [38] used the co-occurrence characteristics of DNS queries to C&C domain names from multiple malware-infected hosts in a network to extend domain name blacklists. Also, Rahbarinia et al. [37] proposed Segugio to detect new C&C domain names from DNS query behaviors in large ISP networks. This system requires malware-infected hosts in a network; however, our approach works without malware-infected hosts.

Bilge et al. [8] proposed Exposure, which detects malicious domain names on the basis of the time series changes in the number of DNS queries in passive DNS data. Perdisci et al. [35] proposed FluxBuster, which detects previously unknown fast-flux domain names by using large-scale passive DNS data. The cost of Exposure or FluxBuster for retrieving and analyzing large-scale passive DNS logs is much larger than that of our TVPs in DOMAINPROFILER.

Antonakakis et al. [6] proposed Kopis, which uses user behavior observed in passive DNS logs on authoritative DNS servers. Today, the number of new generic top-level domains (gTLDs) is rapidly increasing; thus, it is more difficult to exhaustively gather such information on a TLD's authoritative DNS servers. DOMAINPROFILER does not require such logs and is designed to use publicly available information.

Antonakakis et al. [7] also proposed Pleiades, which is focused on DNS queries to nonexistent domain names observed on recursive DNS servers to detect DGA domain names used for C&C. In addition, Thomas and Mohaisen proposed a system similar to Pleiades to determine the characteristics of nonexistent domain names [42]. Our system does not require such DNS logs and is focused on not only C&C domain names but also other malicious domain names such as drive-by download and phishing.

6.3 Historic relationship approach

The historic relationship approach is focused on the historic or time series information of domain names, IP addresses, and web content.

Antonakakis et al. [5] proposed a system called Notos to detect malicious domain names that have similar patterns to past malicious domain names. This was one of the most successful studies on domain name evaluation or reputation systems. Notos uses historic IP addresses and historic domain names to extract effective features to discriminate malicious domain names from legitimate ones. As stated in Sect. 3.2.2, we use these features as some of our features in related IP addresses (rIPs) and related domain names (rDomains). Moreover, our TVP features dramatically expand detection and prediction performance, as discussed in Sect. 4.

Manadhata et al. [31] proposed a method for detecting malicious domain names from event logs in an enterprise network by using graph-based analysis. Boukhtouta et al. [9] proposed an analysis method for creating graphs from sandbox results to understand the relationships among domain names, IP addresses, and malware family names. Kührer et al. [24] proposed a method for identifying parked and sinkhole domain names from Web sites and blacklist content information by using graph analysis. DOMAINPROFILER strongly relies on the TVP or time series information, which these studies did not use, to precisely predict future malicious domain names. Chiba et al. used the characteristics of past

malicious IP addresses to detect malicious Web sites [14]. Our system uses not only IP address features (rIPs) but also TVPs to precisely detect malicious domain names.

Venkataraman et al. [43] developed a method for inferring time series shifting of IP address prefixes to detect malicious IP addresses used for spam or botnet. DOMAINPROFILER is also focused on the idea of shifting malicious resources; however, the target and method are completely different.

The closest concept to ours is that proposed by Soska and Christin [40]. They focused on the idea of variations in compromised Web sites using a popular content management system and proposed a method for predicting vulnerable Web sites before they turn malicious. The main features they rely on are content-based features obtained from compromised Web sites. The concept of DOMAINPROFILER seems to be similar; however, our system has an advantage in scalability because it does not need to access Web sites or extract features from them. Moreover, the focus with our system is wider; that is, DOMAINPROFILER can detect Web sites related to drive-by download and phishing attacks.

Lever et al. [25] pointed out a problem in re-registration of expired domain names and developed an algorithm called Alembic to find potential domain name ownership changes using passive DNS data. We have also focused on the temporal changes in domain names including such re-registered domain names [15]. However, our system does not rely on passive DNS data, and its goal is not only finding re-registered domain names but also specifying truly malicious domain names abused by attackers.

Recently, Hao et al. [21] proposed predator, which predicts future malicious domain names when they are registered. Their system uses domain registration information directly obtained from the .com TLD registry (VeriSign, Inc). However, more and more new gTLDs (e.g., .xyz and .top) have started being used since October 2013. The number of such new gTLDs was 1,184 as of September 2016 [22]. Attackers also leverage new gTLDs for their cyber-attacks. For example, Halvorson et al. [20] showed that domain names using new gTLDs are twice as likely to appear on blacklists; this means attackers now actively make use of new gTLDs. Obviously, to keep up with such situations, predator needs to obtain real-time access privileges to highly confidential data inside the each new gTLD's registry. Although the concept of predator resembles that of DOMAINPROFILER, their mechanisms are totally different because our system does not require any data only owned by a registrar, registry, and authoritative name server.

7 Conclusion

We proposed DOMAINPROFILER to detect/predict domain names that will potentially be maliciously used in future.

Our key idea behind the system is to exploit temporal variation patterns (TVPs) of malicious domain names. A TVP of domain names includes information about how and when a domain name has been listed in legitimate/popular and/or malicious domain name lists. Our system actively collects historical DNS logs, identifies their TVPs, and predicts whether a given domain name will be used maliciously. Our evaluation with large-scale data revealed that DOMAINPROFILER can predict malicious domain names 220 days beforehand with a true positive rate (TPR) of 0.985. Moreover, we verified the effectiveness of our system in terms of the benefits from our TVPs and defense against cyber-attacks. DOMAINPROFILER will be one way to track the trend in ever-changing cyber security threats.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- (2017) Alexa top sites. URL <http://www.alexa.com/topsites>
- (2017) hpHosts. URL <http://www.hosts-file.net/>
- (2017) VirusTotal. URL <https://www.virustotal.com/>
- Alrwais, S.A., Yuan, K., Alowaisheq, E., Li, Z., Wang, X.: Understanding the dark side of domain parking. In: Proceedings of the 23rd USENIX security symposium, pp. 207–222 (2014)
- Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a dynamic reputation system for DNS. In: Proceedings of the 19th USENIX security symposium, (2010)
- Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou, N. II, Dagon, D.: Detecting malware domains at the upper DNS hierarchy. In: Proceedings of the 20th USENIX security symposium, (2011)
- Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., Dagon, D.: From throw-away traffic to bots: detecting the rise of DGA-based malware. In: Proceedings of the 21st USENIX security symposium, (2012)
- Bilge, L., Sen, S., Balzarotti, D., Kirda, E., Kruegel, C.: Exposure: a passive DNS analysis service to detect and report malicious domains. *ACM Trans. Inf. Syst. Secur.* **16**(4), 14:1–14:28 (2014). <https://doi.org/10.1145/2584679>
- Boukhtouta, A., Mouheb, D., Debbabi, M., Alfandi, O., Iqbal, F., Barachi, M.E.: Graph-theoretic characterization of cyber-threat infrastructures. *Digit. Invest.* **14**(Supplement 1), S3–S15 (2015). <https://doi.org/10.1016/j.diin.2015.05.002>
- Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
- Breiman, L.: Manual on setting up, using, and understanding random forests v3.1. (2002) URL https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf
- CAIDA (2017) Routeviews prefix to AS mappings dataset (pfx2as) for IPv4 and IPv6. URL <http://www.caida.org/data/routing/routeviews-prefix2as.xml>
- Chen, Y., Antonakakis, M., Perdisci, R., Nadji, Y., Dagon, D., Lee, W.: DNS noise: Measuring the pervasiveness of disposable domains in modern DNS traffic. In: Proceedings of the 44th annual IEEE/IFIP international conference on dependable systems and

- networks (DSN'14), pp 598–609, (2014). <https://doi.org/10.1109/DSN.2014.61>
14. Chiba, D., Tobe, K., Mori, T., Goto, S.: Detecting malicious websites by learning IP address features. In: Proceedings of the 12th IEEE/IPSJ international symposium on applications and the internet (SAINT'12), pp 29–39, (2012). <https://doi.org/10.1109/SAINT.2012.14>
 15. Chiba, D., Yagi, T., Akiyama, M., Shibahara, T., Yada, T., Mori, T., Goto, S.: DomainProfiler: Discovering domain names abused in future. In: Proceedings of the 46th annual IEEE/IFIP international conference on dependable systems and networks (DSN'16), pp 491–502, (2016). <https://doi.org/10.1109/DSN.2016.51>
 16. Eshete, B., Alhuzali, A., Monshizadeh, M., Porras, P.A., Venkatakrishnan, V.N., Yegneswaran, V.: EKHunter: A counter-offensive toolkit for exploit kit infiltration. In: Proceedings of the 22nd annual network and distributed system security symposium (NDSS'15), (2015)
 17. Farsight Security, Inc (2017) DNSDB. URL <https://www.dnsdb.info/>
 18. Felegyhazi, M., Kreibich, C., Paxson, V.: On the potential of proactive domain blacklisting. In: Proceedings of the 3rd USENIX conference on large-scale exploits and emergent threats (LEET'10), (2010)
 19. Grier, C., Ballard, L., Caballero, J., Chachra, N., Dietrich, C.J., Levchenko, K., Mavrommatis, P., McCoy, D., Nappa, A., Pitsilidis, A., Provos, N., Rafique, M.Z., Rajab, M.A., Rossow, C., Thomas, K., Paxson, V., Savage, S., Voelker, G.M.: Manufacturing compromise: the emergence of exploit-as-a-service. In: Proceedings of the 19th ACM conference on computer and communications security (CCS'12), pp 821–832, (2012). <https://doi.org/10.1145/2382196.2382283>
 20. Halvorsen, T., Der, M.F., Foster, I., Savage, S., Saul, L.K., Voelker, G.M.: From .academy to .zone: An analysis of the new TLD land rush. In: Proceedings of the ACM conference on internet measurement conference (IMC'15), pp 381–394, (2015). <https://doi.org/10.1145/2815675.2815696>
 21. Hao, S., Kantchelian, A., Miller, B., Paxson, V., Feamster, N.: PREDATOR: Proactive recognition and elimination of domain abuse at time-of-registration. In: Proceedings of the 23rd ACM conference on computer and communications security (CCS'16), (2016)
 22. ICANN (2017) ICANN new gTLDs delegated strings. URL <https://newgtlds.icann.org/en/program-status/delegated-strings>
 23. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. *J. Am. Stat. Assoc.* **53**(282), 457–481 (1958)
 24. Kühner, M., Rossow, C., Holz, T.: Paint it black: Evaluating the effectiveness of malware blacklists. In: Proceedings of the 17th international symposium on research in attacks, intrusions and defenses (RAID'14), pp 1–21, (2014). https://doi.org/10.1007/978-3-319-11379-1_1
 25. Lever, C., Walls, R.J., Nadji, Y., Dagon, D., McDaniel, P., Antonakakis, M.: Domain-Z: 28 registrations later measuring the exploitation of residual trust in domains. In: Proceedings of the 37th IEEE symposium on security and privacy (SP'16), pp 691–706, (2016). <https://doi.org/10.1109/SP.2016.47>
 26. Li, Z., Alrwais, S.A., Xie, Y., Yu, F., Wang, X.: Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In: Proceedings of the 34th IEEE symposium on security and privacy (SP'13), pp 112–126, (2013). <https://doi.org/10.1109/SP.2013.18>
 27. Liaw, A., Wiener, M.: Classification and regression by randomForest. *R News* **2**(3):18–22 (2002) URL <http://CRAN.R-project.org/doc/Rnews/>
 28. Liaw, A., Wiener, M.: Package 'randomForest'. (2015) URL <https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>
 29. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'09), pp 1245–1254, (2009). <https://doi.org/10.1145/1557019.1557153>
 30. Maigron, P.: Regional internet registries statistics. (2017) URL http://www-public.tem-tsp.eu/~maigron/RIR_Stats/index.html
 31. Manadhata, P., Yadav, S., Rao, P., Horne, W.: Detecting malicious domains via graph inference. In: Computer security—ESORICS 2014, lecture notes in computer science, vol 8712, Springer, Berlin, pp 1–18, (2014). https://doi.org/10.1007/978-3-319-11203-9_1
 32. MaxMind (2017) GeoIP2 databases. URL <https://www.maxmind.com/en/geoip2-databases>
 33. Mockapetris, P.: Domain names - implementation and specification. RFC 1035, (1987) URL <http://www.ietf.org/rfc/rfc1035.txt>
 34. Mozilla foundation: Public suffix list. (2017) URL <https://publicsuffix.org/list/>
 35. Perdisci, R., Corona, I., Giacinto, G.: Early detection of malicious flux networks via large-scale passive DNS traffic analysis. *IEEE Trans. Depend. Sec. Comput.* **9**(5), 714–726 (2012). <https://doi.org/10.1109/TDSC.2012.35>
 36. Rahbarinia, B., Perdisci, R., Antonakakis, M., Dagon, D.: SinkMiner: Mining botnet sinkholes for fun and profit. In: Proceedings of the 6th USENIX workshop on large-scale exploits and emergent threats (LEET'13), (2013)
 37. Rahbarinia, B., Perdisci, R., Antonakakis, M.: 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, (2015). <https://doi.org/10.1109/DSN.2015.35>
 38. Sato, K., Ishibashi, K., Toyono, T., Miyake, N.: Extending black domain name list by using co-occurrence relation between DNS queries. In: Proceedings of the 3rd USENIX conference on large-scale exploits and emergent threats (LEET'10), (2010)
 39. Sebastián, M., Rivera, R., Kotzias, P., Caballero, J.: AVclass: A tool for massive malware labeling. In: Proceedings of the 19th international symposium on research in attacks, intrusions, and defenses (RAID'16), pp 230–253 (2016). https://doi.org/10.1007/978-3-319-45719-2_11
 40. Soska, K., Christin, N.: Automatically detecting vulnerable web-sites before they turn malicious. In: Proceedings of the 23rd USENIX security symposium, pp 625–640, (2014)
 41. Szurdi, J., Kocso, B., Cseh, G., Spring, J., Felegyhazi, M., Kanich, C.: The long “taile” of typosquatting domain names. In: Proceedings of the 23rd USENIX security symposium, pp 191–206, (2014)
 42. Thomas, M., Mohaisen, A.: Kindred domains: Detecting and clustering botnet domains using DNS traffic. In: Proceedings of the 23rd international conference on World Wide Web (WWW'14), pp 707–712, (2014). <https://doi.org/10.1145/2567948.2579359>
 43. Venkataraman, S., Brumley, D., Sen, S., Spatscheck, O.: Automatically inferring the evolution of malicious activity on the Internet. In: Proceedings of the 20th annual network and distributed system security symposium, (NDSS'13) (2013)
 44. Verisign (2017) Domain name industry brief. URL http://www.verisign.com/en_US/innovation/dnib/index.xhtml
 45. Vissers, T., Joosen, W., Nikiforakis, N.: Parking sensors: Analyzing and detecting parked domains. In: Proceedings of the 22nd annual network and distributed system security symposium, (NDSS'15), (2015)
 46. Weimer, F.: Passive DNS replication. In: Proceedings of the 17th annual FIRST conference, (2005)
 47. Yadav, S., Reddy, A.K.K., Reddy, A.L.N., Ranjan, S.: Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. *IEEE/ACM Trans. Netw.* **20**(5), 1663–1677 (2012). <https://doi.org/10.1109/TNET.2012.2184552>