Don't Let One Rotten Apple Spoil the Whole Barrel: Towards Automated Detection of Shadowed Domains

Daiping Liu
University of Delaware
dpliu@udel.edu

Zhou Li
ACM Member
lzcarl@gmail.com

Kun Du
Tsinghua University
dk15@tsinghua.edu.cn

Haining Wang
University of Delaware
hnw@udel.edu

Baojun Liu
Tsinghua University
lbj15@mails.tsinghua.edu.cn

Haixin Duan
Tsinghua University
duanhx@tsinghua.edu.cn

ABSTRACT

Domain names have been exploited for illicit online activities for decades. In the past, miscreants mostly registered new domains for their attacks. However, the domains registered for malicious purposes can be deterred by existing reputation and blacklisting systems. In response to the arms race, miscreants have recently adopted a new strategy, called *domain shadowing*, to build their attack infrastructures. Specifically, instead of registering new domains, miscreants are beginning to compromise legitimate ones and spawn malicious subdomains under them. This has rendered almost all existing countermeasures ineffective and fragile because subdomains inherit the trust of their apex domains, and attackers can virtually spawn an infinite number of shadowed domains.

In this paper, we conduct the first study to understand and detect this emerging threat. Bootstrapped with a set of manually confirmed shadowed domains, we identify a set of novel features that uniquely characterize domain shadowing by analyzing the deviation from their apex domains and the correlation among different apex domains. Building upon these features, we train a classifier and apply it to detect shadowed domains on the daily feeds of VirusTotal, a large open security scanning service. Our study highlights domain shadowing as an increasingly rampant threat. Moreover, while previously confirmed domain shadowing campaigns are exclusively involved in exploit kits, we reveal that they are also widely exploited for phishing attacks. Finally, we observe that instead of algorithmically generating subdomain names, several domain shadowing cases exploit the wildcard DNS records.

1 INTRODUCTION

The domain name system (DNS) serves as one of the most fundamental Internet components and provides critical naming services for mapping domain names to IP addresses. Unfortunately, it has also been constantly abused by miscreants for illicit online activities. For instance, botnets exploit algorithmically generated domains to circumvent the take-down efforts of authorities [11, 65, 86], and scammers set up phishing websites on domains resembling well-known legitimate ones [38, 75]. In the past, Internet miscreants mostly registered new domains to launch attacks. To mitigate the threats, tremendous efforts [10, 14, 33, 41, 77] have been devoted in the last decade to construct reputation and blacklisting systems that can fend off malicious domains before visited by users. All of these endeavors render it less effective to register new domains for attacks. In response, miscreants have moved forward to more sophisticated and stealthy strategies.

In fact, there is a newly emerging class of attacks adopted by cybercriminals to build their infrastructure for illicit online activities, *domain shadowing*, where instead of registering new domains, miscreants infiltrate the registrant accounts of legitimate domains and spawn subdomains under them for malicious purposes. Domain shadowing is becoming increasingly popular due to its superior ability to evade detection. The shadowed domains naturally inherit the trust of a legitimate parent zone, and miscreants can even set up authentic HTTPS connections with Let's Encrypt [59]. Even worse, miscreants can create an infinite number of subdomains under many hijacked legitimate domains and rapidly rotate among them at no cost. This makes it quite challenging to keep blacklists up-to-date and gather useful information for meaningful analysis. While domain shadowing has been reported in public outlets like `blogs.cisco.com` [8, 40], most previous studies only elaborate on sporadic cases collected in a short time through manual analysis. It is still unclear how serious the threat is and how to address this domain shadowing problem on a larger scale.

In this paper, we conduct the first comprehensive study of domain shadowing in the wild, and we present a novel system to automatically detect shadowed domains by addressing the following unique challenges. Shadowed domains by design do not present suspicious registration information, and thus all detectors leveraging these data [30, 33, 34] can be easily bypassed. Blindly blacklisting all sibling subdomains of shadowed domains is also infeasible in practice, since it can cause large amounts of collateral damage. Last but not least, most suspicious DNS patterns identified in previous studies do not work well in domain shadowing. For instance, Kopis [10] analyzes the collective features of all visitors to a domain. However, our study has seen many shadowed domains being visited only once, rendering the collective features insignificant. Such collective features can be applied to malicious apex domains[1] because

CCS '17, October 30-November 3, 2017, Dallas, TX,USA

ACM ISBN 978-1-4503-4946-8/17/10…$15.00
https://doi.org/10.1145/3133956.3134049

[1] An apex domain is also known as a bare/base/naked/root domain that is separated from the top level domain by a dot, e.g., foo.com, and needs to be purchased from registrars.

the domain registration cost will become unaffordable if an apex is used only a few times.

To bootstrap the design of our detector, we collect a set of 26,132 confirmed shadowed domains under 4,862 distinct zones through manually searching and reviewing technical reports by security professionals. Comparing them with legitimate subdomains, we find that the shadowed ones can be characterized and distinguished by two dimensions. On one hand, shadowed domains usually exhibit deviant behaviors and are more isolated from those known-good subdomains under the same parent zone. For instance, most legitimate domains are hosted on reputable servers, which usually strictly restrict illicit content. Due to the nature of their criminal activity and their demand to evade detection and possible take-down, shadowed domains have to be hosted on cheap and cybercriminal-friendly servers. This deviation serves as a prominent indicator of potential shadowed domains. On the other hand, miscreants tend to exploit a set of shadowed domains under different parent zones within the same campaign. This can greatly increase the resilience and stealthiness of their infrastructure. However, such correlation also presents suspicious synchronous characteristics. For instance, shadowed domains in the same campaign usually appear and disappear at the same time.

Based on these observations, we develop a novel system, called `Woodpecker`, to automatically detect shadowed domains by inspecting the deviation of subdomains to their parent zones and the correlation of shadowed domains among different zones. In particular, we compose 17 features characterizing the usage, hosting, activity, and name patterns of subdomains, based on the passive DNS data. Five classifiers (Support Vector Machine, RandomForest, Logistic Regression, Naive Bayes, and Neural Network) are then trained using these features. We achieve a 98.5% detection rate with an approximately 0.1% false positive rate with a 10-fold cross-validation when using RandomForest.

`Woodpecker` is envisioned to be deployed in several scenarios, e.g., domain registrars and upper DNS hierarchy as a complement to Kopis [10], generating more accurate indicators about the ongoing cybercrimes. In this paper, we demonstrate a use case in which `Woodpecker` is deployed on the open security service VirusTotal (VT) [82]. Specifically, we run our trained classifier over a large-scale dataset built using all subdomains submitted to VirusTotal [82] during February~April 2017 as seeds. The dataset contains 22,481,892 unique subdomains under 2,573,196 parent zones. These domains are hosted on 4,809,728 IP addresses.

Our findings. Applying `Woodpecker` to the daily feeds of VirusTotal, we obtain 287,780 reports, of which 127,561 are confirmed as shadowed domains with a set of heuristics (most of the remaining ones are about malicious apex domains). Our measurement of the characteristics of these shadowed domains indicates that they exhibit quite different properties from conventional malicious domains, and thus existing systems can hardly detect the shadowed domains. Our manual assessment of the security measures of domain registrars shows that their current practices cannot effectively protect the users. We also observe two interesting cases in our results. First, shadowed domains currently exposed in the technical blogs are exclusively involved in exploit kits. However, our detection results show that shadowed domains are also widely exploited in phishing attacks. Another interesting finding is that miscreants also exploit the wildcard DNS records to spawn shadowed domains.

Roadmap. The remainder of this paper is organized as follows. Section 2 introduces the background of DNS and shadowed domains. Section 3 presents the design and extracted features of our detector. In Section 4, we validate the efficacy of our detector using labeled datasets. We then conduct a large-scale analysis of the shadowed domains in Section 5. Section 6 discusses the limitations of our detection approach. Finally, we survey related work in Section 7 and conclude in Section 8.

2 BACKGROUND

We give a brief overview of the domain system in the beginning of this section. Then, we describe the schema regarding domain shadowing attacks and use one real-world case identified by our detection system to walk through the attack flow.

2.1 Basics of Domain Names

Domain name structure. A domain name is presented in the structure of hierarchical tree (e.g., `a.example.com`), with each level (e.g., `example.com`) associated with a DNS zone. For one DNS zone, there is a single manager that oversees the changes of domains within its territory and provides authoritative name-to-address resolution services through the DNS server. The top of the domain hierarchy is the root zone, which is usually represented as a dot. So far, the root zone is managed by ICANN, and there are 13 logical root servers operated by 12 organizations. Below the root level is the top-level domain (TLD), a label after the rightmost dot in the domain name. The commonly used TLDs are divided into three groups, including generic TLDs (gTLDs) like `.com`, country-code TLDs (ccTLDs) like `.uk`, and sponsored TLDs (sTLDs) like `.jobs`. Next to TLD is the second-level domain (2LD) (e.g., `.example.com`), which can be directly registered from registrars (like GoDaddy) if not yet occupied, in most cases. One exception occurs when both ccTLD and gTLD appear in the domain name, like `.co.uk`, and the registrants must choose a 3rd-level domain (3LD), like `example.co.uk`. In this work, we use *effective TLDs* (eTLDs) or *public suffix* to refer to the TLDs directly operated by registrars (like `.com` and `.co.uk`), and *apex domains* (or *apex* in short) to refer to domains that can be obtained under eTLDs. The registrant that owns the apex domain is allowed to create *subdomains*, like 3LDs and 4LDs without asking permission from the registrar. In the meantime, the registrant takes responsibility for managing the domain resolution, either by running her own DNS server or using other public DNS servers.

DNS record. When a registrant requests a domain name from a registrar, the request is also forwarded to a registry (e.g., Verisign), which controls the domain space under the eTLD and publishes DNS records (or *resource records (RR)*) in the zone file. Similarly, a subdomain creation request also causes changes in the zone file, except that the request can be handled by the owner herself. An RR is a tuple consisting of five fields, <*name*, *TTL*, *class*, *type*, *data*>, where *name* is a fully qualified domain name (FQDN), *TTL* specifies the lifetime in seconds of a cached RR, *class* is rarely used and is almost always `"IN"`, *type* indicates the format of a record, and *data* is the record-specific data, e.g., an IP address of a domain.

Figure 1: Adding a subdomain in domain registrar GoDaddy. Assume the apex domain is foo.com. The added subdomain is `shadowed.foo.com`.

Subdomain management. Domain owners can create and manage subdomains under their apex domains through web GUI or API provided by registrars. There are three types of DNS RRs associated with subdomain creation. An `A` record maps a domain name to an IPv4 address, e.g., `foo.example.com A 1.1.1.1`. A `CNAME` record specifies the alias of a canonical domain, e.g., `foo.example.com CNAME bar.another.com`. An `AAAA` record maps a domain name to an IPv6 address, e.g., `foo.example.com AAAA 0:0:0:0:0:0:0:1`. Figure 1 shows the web interface of GoDaddy for subdomain creation. Assume the apex is `foo.com` and the `Host` field is filled with `shadowed`. A new subdomain `shadowed.foo.com` will be created after the submission of request, which updates the zone file shortly. The domain owner could fill the `Host` field with `*` to create a *wildcard record*. As a result, any request to the non-existent subdomain (not specified by an `A`, `AAAA` or `CNAME` record) will be captured and resolved to the corresponding IP.

2.2 Domain Shadowing

A malicious web host is a critical asset in the cybercriminal infrastructure. To prevent hosts from being easily discovered, like exposing their physical existence from IPs, attackers abuse DNS services and hide the hosts behind the ever-changing domain names. Many attackers choose to own domain names from registrars. Since malicious domains are ephemeral, usually revoked shortly after being detected, they prefer to register many domains at a low price and short expiration duration. This strategy, however, leaves the malicious domains more distinguishable from the legitimate domains when examined by domain reputation systems [14, 30, 33, 57, 58].

Recently, attackers have begun to compromise the domain system to evade existing detection systems while confining the cost of obtaining domains. Discovered by Cisco Talos in 2015 [40], Angler, an exploit kit with widespread usage by underground actors, evolved its infrastructure and used the subdomains under the legitimate domains as redirectors to cover the exploit servers. In particular, the bad actors harvested a large amount of credentials of domain owners (e.g., through phishing emails or brute-force guessing) and logged into their accounts to create subdomains. This technique is called *domain shadowing*, and such subdomains are called *shadowed domains*.

Domain shadowing is quite effective in evading existing detection systems for several reasons. First, many registrants use weak passwords and never check the domain configuration after its creation [23]. In addition, the changes are not submitted to the registries' zone file, setting aside the monitoring system of registries. Second, there is usually little restriction over subdomain creation. As long as a domain consists of less than 127 levels and the name length is less than 253 ASCII characters, the domain name is valid.

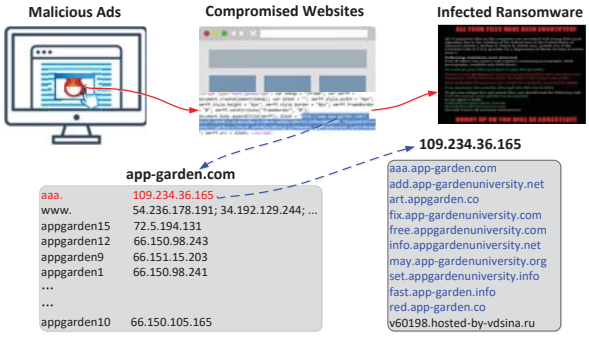


Figure 2: Shadowed domains used in a campaign of EITest Rig EK in April 2017. `app-garden.com` is a legitimate apex domain.

This leaves virtually infinite space for an adversary to rotate domains and evade blacklists. Third, the malicious subdomains inherit the reputation of legitimate apex domains. As information from the Whois record could greatly impact the domain score outputted by many systems [33] and subdomains share the same values as their apex domains, the shadowed domains can easily slip through the existing detection systems.

In addition to compromising registrant credentials, vulnerabilities in registrars and DNS servers could also lead to domain shadowing. For instance, it has been reported that several reputable registrars were breached and massive domain credentials were leaked, including `Name.com` [67], `punto.pe` [27], and `Hover` [79]. As a result, malicious subdomains could be created under a large volume of apex domains at the same time. Moreover, the zone files hosted by the authoritative DNS servers could be targeted by domain hackers who manipulate the RR data to change or add domains [44].

Scope. In this work, we aim to detect shadowed domains *created in bulk* by domain hackers. While the existing research revealed that this technique was mainly used by exploit kits (see the description of our ground-truth data in Table 2), we consider all attacks leveraging this technique, like phishing, in our study. Changing and deleting subdomains without the owners' consent, which could achieve the same goal or cause service interruption, are not considered in this paper, given that they are less likely to be used and observed. While subdomains could be created under malicious apex domains, they are not the focus of our study and could be handled by existing tools gauging domain reputation like PREDATOR [33]. Targeted attacks like APT (Advanced Persistent Threat) operate on a small number of domains, including subdomains under legitimate apex domains. Detecting targeted attacks automatically is still a paramount challenge for the security community [29], due to its nominal signal overwhelmed by a large amount of data. We do not expect individual subdomains in these cases to be effectively detected by our system and leave that research as a future direction.

2.3 Real-world Example

Here we demonstrate how domain shadowing empowers attackers' operations using a real-world case recently discovered by our system (illustrated in Figure 2). In this case, we found the

shadowed domains in the passive DNS data (our dataset is described in Section 3.2), and the appearance of the shadowed domains was also documented in a security website [60]. One such domain is `aaa.app-garden.com`, created under a legitimate 2LD `app-garden.com`, which redirects users' traffic from compromised doorway sites to *Rig Exploit Kit (EK)* [72], aiding a malware distribution campaign called *EITest*. In particular, the doorway sites serve malicious advertisements created by attackers, and the JavaScript code redirects the visitor to a sequence of compromised sites until arriving at `aaa.app-garden.com`, which stores Rig EK's drive-by-download code. If the malicious code executes successfully in a user's browser, a ransomeware will be downloaded to encrypt victim's files.

By inspecting the data relevant to the shadowed domains, we discovered several unique features about such an attack. The shadowed domain `aaa.app-garden.com` points to an IP address that is quite different from the apex `app-garden.com` and other sibling subdomains, like `www.app-garden.com`. More specifically, the shadowed domain is associated with an IP in Russia while all other subdomains are linked to IPs in the United States. By inspecting the domains linked to `109.234.36.165` (10 in total from our data), we found that nine of them share similar apex names to `app-garden.com` (e.g., `app-garden.co`). Notably, all nine apex domains were registered by Cook Consulting, Inc., with one in April 2011, six in May 2014, and two in March 2017[2]. We speculate that the domain hacker obtained the login credential and injected the subdomain into many apex domains under the victim's account. It is also interesting that meaningful single words, like `info` and `free`, are used to construct the malicious subdomains. As such, detectors based on random domain names, like DGA detector [11, 86], have a high probability of being evaded.

3 AUTOMATIC DETECTION OF SHADOWED DOMAINS

In response to the emerging threat of domain shadowing, in this section we present our design of an automated detection system, `Woodpecker`. We first overview its workflow and deployment scenarios. Then, we describe the dataset used for training and testing. Finally, we elaborate on the features we use to distinguish shadowed and legitimate domains.

3.1 Overview

We could follow conventional approaches, like content or URL analysis, to detect shadowed domains. However, after our initial exploration, we found that these approaches are not suitable. Many shadowed domains are used as redirectors. Finding the gateways, e.g., compromised sites, is a non-trivial task. Even if we are able to find the shadowed domains and download the content, we may still fail to classify them correctly when they only serve seemingly benign redirection code. Compared to domains owned by attackers, the registration information of a shadowed domain is identical to that of the benign apex domain, which undermines the effectiveness of many approaches based on domain registration.

[2]The `app-garden.com` site was registered through a domain proxy, and the registrant information is not available through the Whois query. However, that domain was registered at the same time as one of the nine domains.

Users' visits to shadowed domains would be observed by DNS servers and further collected by a passive DNS (PDNS) database. Erasing the traces from DNS servers and PDNS is considerably more difficult than compromising websites and domain accounts. As such, we decide to analyze the DNS data to solve our problem. Though the information underlying the DNS data is much more scarce than web content, it is still sufficient to distinguish shadowed and legitimate domains, due to two key insights. First, shadowed domains serve a different purpose from the legitimate parent domains and sibling subdomains: for instance, they could be associated with IPs far from their parents' and siblings', leading to prominent *deviation*. Second, to make malicious infrastructures resilient to take-down efforts, attackers prefer to play domain fluxing and rotate shadowed domains. In the meantime, the IPs covered by them are limited, leading to abnormal *correlation*, especially when they are under apex domains whose owners have no business relations.

Our detection system, `Woodpecker`, is driven by those two insights and runs a novel deviation and correlation analysis on the PDNS data. It takes three steps to detect shadowed domains. Given a set of subdomains $\mathbb{S}$ observed at a certain vantage point (e.g., enterprise proxy and scanning service), we first build the profiles for each apex of $\mathbb{S}$ using the data retrieved from the PDNS source. Assume an apex $\mathbf{D}$ is represented by a set of tuples:

$$\mathbf{D} = \{ s_i \mid s_i :=< name_i, rrtype, rdata, t_f, t_l, count > \}$$

where $name_i$ is the FQND under D, `rrtype` and `rdata` represent the type (e.g., `A` record) and data (e.g., IP) fields within the answers returned by DNS servers , t_f and t_l denote the time when an individual `rdata` is first and last seen, and `count` is the number of DNS queries that receive the `rdata` in response.

In the second step, `Woodpecker` aggregates these profiles and characterizes the subdomains using a set of 17 significant features from the dimensions of deviation and correlation. In addition to the data from PDNS, we also query a public repository of web crawl data to measure the connectivity of domains (only extracting web links). Finally, a machine-learning classifier is trained over a labeled dataset and is further applied to large unlabeled datasets to detect shadowed domains. Figure 3 depicts the workflow of `Woodpecker`.

Deployment. `Woodpecker` is a lightweight detector against shadowed domains, which only requires passive DNS and publicly crawled data. We envision `Woodpecker` to be deployed in several scenarios. It can help domain registrars like GoDaddy to detect domains whose subdomains are added in an unexpected way, and hence allows them to notify domain owners promptly. The operators of DNS servers can deploy our system to trace and mitigate Internet threats. The administrators of organizational networks can use the output of our system to amend their blocked lists (i.e., whether to block a subdomain or an apex domain). Finally, it can be deployed by public scanning services, like VirusTotal [82], to analyze submitted URLs/domains and provide more accurate labels. When these services are used as blacklists and a site is blocked, knowing the label is essential for the owner to diagnose the root cause [15].

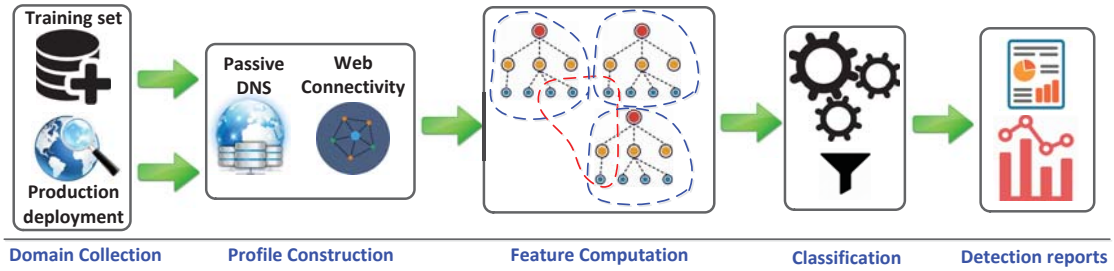


Figure 3: Workflow of Woodpecker.

3.2 Dataset

To bootstrap our study, we collected domains from different sources and queried the PDNS data to build profiles. Below we describe how these data were collected and summarize them in Table 1.

Shadowed domains. Obtaining a list of shadowed domains requires a lot of manual effort. While there are many public blacklists documenting malicious domains, we have not found any such list for shadowed domains specifically. Hence, we rely on web search[3] (using keywords like "domain shadowing" and "shadowed domain") to find all relevant articles. After manually reviewing that information, the indicators (i.e., malicious domains/IPs/hashes) in the articles are downloaded. The subdomains hosted under known malicious apex domains and directly under third-party hosting services are removed for dataset sanitization. Overall, we managed to collect 26,132 known shadowed domains under 4,862 apex domains, as listed in Table 1[4]. Table 2 summarizes this dataset, and we name it $D_{shadowed}$. While all shadowed domains in $D_{shadowed}$ are used for exploit kits, we are able to discover other types of usage, like phishing, from the testing dataset (elaborated in §5.1).

Legitimate domains. We collected legitimate domains as another source to train the classifier. The data comes from two channels. First, we chose domains that are consistently ranked among the top 20,000 from 2014 to 2017 by Alexa [61], and we obtained 8,719 2LDs in total. These popular domains usually have many subdomains that cover a broad spectrum of services, including web, mail, and file downloading. Solely relying on popular domains can introduce bias to our system, so we also obtained non-popular legitimate domains from a one-week DNS trace collected from a campus network. The DNS trace was anonymized and desensitized for our usage. We scanned these domains using VirusTotal and excluded all of the malicious ones (alarmed by at least one participating blacklist). Further, we randomly sampled 2,500 2LDs that were ranked below 500,000 by Alexa in 2017. The two datasets are denoted as D_{pop} and D_{nonpop}. The volume of subdomains found from our legitimate datasets is not very extensive due to the rate limit placed by the PDNS provider, as described later.

VT daily feeds. We evaluated the trained model based on the data downloaded from VT, as a showcase to demonstrate that

[3]Searching Google and `otx.alienvault.com`, a platform sharing threat intelligence.
[4]We rely on a list documenting the public suffix in domain names to extract the apex [62].

```
## From 360
{"rrname": "eu.account.amazon.com", "rrtype": "A",
"rdata": "52.94.216.25;", "count": 31188,
"time_first": 1477960509, "time_last": 1494290720}
## From Farsight
{"rrname":"aws.amazon.com.","rrtype":"A",
"rdata":["54.240.255.207"], "count":63,
"time_first":1302981660,"time_last":1318508315}
```

Figure 4: Two sample records for subdomains under `Amazon.com` from 360 and Farsight (field explanation is covered in Section 3.1).

`Woodpecker` can be readily integrated into security services. In particular, we queried for a live feed of reports on all URLs submitted to VT during February~April 2017 on a daily basis. For each submitted subdomain s_i, we queried VT to obtain the domain report and IP report to include additional information for later result validation. All subdomains without IP and apex information were filtered out, in order to reduce unnecessary queries submitted to PDNS. We further excluded subdomains one level under web hosting services and dynamic DNS based on the `category` field from the VT domain report (e.g., "web hosting" and "dynamic DNS"). This dataset is denoted as D_{vt}, which contains 22,481,892 unique subdomains under 2,573,196 apex domains.

Passive DNS data. We queried the PDNS data of two security companies, Farsight Security [26] and 360 Security [64], to obtain aggregated DNS statistics for apex domains in all datasets (we used a wildcard query, like `*.example.com`, to retrieve the data associated with all subdomains of `example.com`), except D_{VT}. We did not query Farsight for D_{VT} due to its daily rate limit. Our account granted by 360 does not have such restrictions and we queried 360 for all apex domains in D_{VT}. Figure 4 shows two sample records from 360 and Farsight.

The columns 6~8 in Table 1 present the obtained data. As shown, different PDNS databases have varying coverage. The evaluation of the impact of different PDNS sources is presented in §4.3. For $D_{shadowed}$, their siblings under the same apex domains might be added by attackers but missed by security companies. It is desirable to determine whether `Woodpecker` can detect new shadowed domains among them. As such, we constructed another dataset $D_{unknown}$, which includes all unlabeled siblings of $D_{shadowed}$.

Dataset	Category	# of Domains	# of Apex	Farsight		360	
				# of Domains	# of IP	# of Domains	# of IP
$D_{shadowed}$	Shadowed	26,132	4,862	21,958	1,188	7,121	965
$D_{unknown}$	Unlabeled siblings of $D_{shadowed}$	-	-	34,586	27,630	8,573	10,609
D_{pop}	Legitimate popular	-	8,719	8,965,818	3,596,441	1,081,112	645,763
D_{nonpop}	Legitimate unpopular	-	2,500	713,154	349,874	80,920	61,507
D_{vt}	Daily feeds from VirusTotal	-	2,573,196	-	-	22,481,892	4,809,728

Table 1: Training and test datasets. Columns 3~4 include all domains we manually collected and thus some cells like those of $D_{unknown}$ do not have data. Columns 5~8 present the number of domains obtained from two PDNS, Farsight and 360, respectively.

Source	Campaign	# Indicators
blogs.cisco.com	Angler [8, 40]	16,580
blog.talosintelligence.com	Neutrino [78], Angler [71, 80], Sundown [20]	9,536
heimdalsecurity.com	Angler [31]	5
blog.malwarebytes.com	Neutrino [70], Angler [2, 81]	9
proofpoint.com	Angler [69]	2
Total		26,132

Table 2: Sources of confirmed domain shadowing.

3.3 Features of Domain Shadowing

`Woodpecker` inspects the PDNS data collected from the global sensor array to detect shadowed domains. Prior to our work, there have been several approaches using PDNS data to detect malicious domains in general, like Notos [9], Exposure [14], and Kopis [10]. However, these systems are not good choices for finding shadowed domains, due to their different features (e.g., ephemeral and readable names) and appearance in many different attack vectors (not only those used by botnet). We provide a detailed comparison in Appendix §A.

By examining the ground-truth set $D_{shadowed}$, we found a set of features unique to shadowed domains, which are essentially divided into two dimensions.

- **Deviation from legitimate domains under the same apex.** How subdomains are created and used differs greatly between legitimate site owners and domain hackers. To name a few, legitimate subdomains tend to be hosted close to the apex, while shadowed domains are hosted by bullet-proof servers with much fewer restrictions whose IP is far from the apex. A site owner usually creates subdomains gradually while shadowed domains are added in bulk around the same time. The homepage of the apex domain (or `www` subdomain) usually contains a link to legitimate subdomains while shadowed domains are isolated, since the registrar and apex website run different systems and compromising them at the same time is much more difficult.

- **Correlation among shadowed domains under a different apex.** Inspecting a single apex is not always effective. On the other hand, shadowed domains under a different apex might be correlated, when an attacker compromises multiple domain accounts and uses all injected subdomains for the same campaign. For instance, shadowed domains under a different apex might be visited around the same time and point to the same IP address, which rarely happens for legitimate subdomains under a different apex.

In the end, we discovered 17 key features for the detection purpose, under four categories: usage, hosting, activity, and name, as listed in Table 3. All features related to deviation can be defined as $D(s_i, \mathbb{S}_{apex(s_i)})$, where $\mathbb{S}_{apex(s_i)}$ represents all known-good domains under the same apex of s_i. Labeling all known-good domains is impractical when processing massive amounts of data. Instead, we simply consider the apex domain and `www` subdomain as known-good. Site owners usually create `www` subdomains for serving web content after the domain is purchased, so they are rarely taken by attackers. The correlation features are extracted from subdomains hosted together, i.e., sharing the same IP. We choose IP to model correlation since legitimate websites tend to avoid sharing the IP with attackers. Below we elaborate the details of each feature.

3.3.1 *Subdomain Usage.*

This category characterizes how subdomains are visited, their popularity and web connectivity.

Days between first non-`www` and apex domain. We check when the first non-`www` subdomain was created under the apex. We found that many compromised apex domains only run websites, whose only legitimate subdomain is a `www` domain. Therefore, a new subdomain created suddenly should be considered suspicious. Assume `Date(d)` is the date when a domain `d` is first seen. We compute this feature as $F1 = \frac{1}{\log(Date(s) - Date(apex(s)) + 1)}$, where `s` is the first non-`www` subdomain under its apex and `apex(s)` denotes its apex. If there are no subdomains or all subdomains are created on the same day as their apex, this feature is set to 1.

Ratio of popular subdomains. Miscreants usually generate names of their shadowed domains algorithmically. We observe that the names tend to avoid being overlapped with popular subdomain names, as changing the existing subdomain is not among the attacker's goals. Based on this observation, we define two features, the ratio of popular subdomains under the upper apex and on an IP[5]. Specifically, given a suspicious subdomain s, we compute $F2 = \frac{|\{POP(d_i)\}|}{|\{d_i | 2LD(d_i) == 2LD(s)\}|}$ and $F3 = \min_{j=1..n}\{\frac{|\{POP(d_i)\}|}{|\{d_i | IP(d_i) == IP_j(s)\}|}\}$, where IP_j is the j^{th} IP of s. For $POP(d_i)$, we only consider subdomains with only one more level than their apex. For example, `www.foo.com` is a popular subdomain under `foo.com` while `www.a.foo.com` is not. We examined the Forward DNS names collected by Project Sonar [25] and selected the top 50 names for popular subdomains, as listed in Table 4.

Web connectivity. Shadowed domains are irrelevant to the services provided by their apex, sibling and hosting servers. As a result, they are not connected to the homepage or other subdomains through web links, while connections between legitimate subdomains and apex are more likely established. Furthermore, a

[5]We issue additional PDNS queries to obtain subdomains not shown in the collected datasets for an uncovered IP.

Category	Feature ID	Feature Name	Dimension	Novel
Subdomain Usage	F1	Days between 1st non-www and apex domain	D	√
	F2	Ratio of popular subdomains under the same apex domain	D	√
	F3	Ratio of popular subdomains co-hosted on the same IP	C	√
	F4	Web connectivity of a subdomains	D	√
	F5	Web connectivity of subdomains under the same apex domains	D	√
	F6	Web connectivity of subdomains co-hosted on the same IP	C	√
Subdomain Hosting	F7	Deviation of a subdomain's hosting IPs	D	√
	F8	Average IP deviation of subdomains co-hosted on the same IP	C	√
	F9	Correlation ratio in terms of co-hosting subdomain number	C	[14]
	F10	Correlation ratio in terms of co-hosting apex number	C	[14]
Subdomain Activity	F11	Distribution of first seen date	C	√
	F12	Distribution of resolution counts among subdomains on the same IP	C	√
	F13	Reciprocal median of resolution counts among subdomains on the same IP	C	√
	F14	Distribution of active days among subdomains on the same IP	C	√
	F15	Reciprocal median of active days among subdomains on the same IP	C	√
Subdomain Name	F16	Diversity of domain levels	C	√
	F17	Subdomain name length	C	[11, 33]

Table 3: Features used in our approach to detect shadowed domains. Feature dimensions D and C denote Deviation and Correlation, respectively. Although some features use the same data source as previous work, e.g., resolution counts as in [4, 51], we model them in different ways.

www	mail	remote	blog	webmail
server	ns1	ns2	smtp	secure
vpn	m	shop	ftp	mail2
test	portal	ns	ww1	host
support	dev	web	bbs	ww42
mx	email	cloud	1	mail1
2	forum	owa	www2	gw
admin	store	mx1	cdn	api
exchange	app	gov	2tty	vps
govyty	hgfgdf	news	1rer	lkjkui

Table 4: List of top 50 popular subdomain names.

shadowed domain is hardly accessible to web crawlers that aim to index web pages, and cloaking is frequently performed.

Here we use the data collected by public web crawlers, including Internet Archive [12] and CommonCrawl [21], to measure the connectivity[6]. For each subdomain s, we issue a query to Internet Archive and CommonCrawl. If any page under s is found to be indexed, this feature, denoted as `F4 = WEB(s)`, is set to 1. Otherwise, it is set to 0.

Additionally, we compute $F5 = \frac{\sum WEB(d_i)}{|\{d_i|2LD(d_i)==2LD(s)\}|}$ and $F6 = \min_{j=1..n}\{\frac{\sum WEB(d_i)}{|\{d_i|IP(d_i)==IP_j(s)\}|}\}$, or the ratio of reachable subdomains under the same apex and same IP. Although accurately assessing connectivity is impossible, we observe that these two crawlers have good coverage of the legitimate domains and hence provide a solid approximation.

3.3.2 Subdomain Hosting.

Deviation of hosting IP. Shadowed domains are usually hosted on IP addresses distant from their apex domain and other known-good subdomains. By contrast, legitimate subdomains tend to be hosted within one region, e.g., within the same autonomous system (AS). Given an apex domain $A = \{< f_i, l_i, ip_i >\}_{i=1..n}$ and its subdomains $S = \{< f_i, l_i, ip_i >\}_{i=1..m}$, where f_i and l_i denote the first and last seen date of ip_i, the deviation (F7) is computed as,

$$Dev(A,S) = max_{j=1..m}\{min_{i=1..n}\{\psi(A_i,S_j)|A(f_i) < S(f_j)\}\} \quad (1)$$

[6]We did not query search engines like Google, because queries are blocked when sending too many.

where $\psi(A_i, S_j)$ is a function that computes the deviation score between two IP records. It is defined as,

$$\psi(A_i,S_j) = \sum_{C\in\{IP,ASN,CC\}} w_k(C[A_i] \neq C[S_j]) \quad (2)$$

where w_k is the weighted penalty for the binary difference between A_i and S_j in IP, AS number (ASN), and country code (CC). We empirically set the weights to 0.3, 0.2, and 0.5. For example, if A_i and S_j share the same IP, the deviation score is 0 (ASN and CC are identical, too). Otherwise, if A_i and S_j share the same ASN but not the same IP, the deviation score will be 0.3. If all of these attributes are different, the deviation score reaches 1.0. Additionally, we compute the average deviation ($F8$) of all subdomains hosted on the same IP.

Correlation ratio. In order to characterize the co-hosting properties of subdomains, we define two features. First, given a subdomain $s = \{IP_j\}_{j=1..n}$, we compute how many subdomains are co-hosted with s, specifically $F9 = \min_{j=1..n}\{\frac{1}{\log(|\{d_i|IP_j(d_i)==IP_j(s)\}|+1)}\}$. This feature alone cannot distinguish shadowed and legitimate subdomains, as we found that some IPs are hosting tens of thousands of legitimate subdomains, probably used by CDN. To address this issue, we count the distinct apex whose subdomains are hosted together with s. The reason behind using this feature is that most site owners prefer to have a dedicated host with a dedicated IP after we filter out the domains that belong to shared hosting and dynamic DNS. We compute $F10 = \min_{j=1..n}\{\frac{1}{\log(|\{2LD(d_i)|IP_j(d_i)==IP_j(s)\}|+1)}\}$ for this feature.

Take the case described in §2.3 as an example to explain how the feature values are computed. There are 11 subdomains from 11 distinct 2LDs co-hosted with `aaa.app-garden.com`. Therefore, the two feature values are $(\frac{1}{\log 12}, \frac{1}{\log 12})$. By contrast, legitimate subdomains under `app-garden.com`, like `appgarden15.app-garden.com`, do not co-host with any other subdomains, and their feature values are $(\frac{1}{\log 2}, \frac{1}{\log 2})$.

3.3.3 Subdomain Activity.

To evade blacklists, miscreants tend to create many shadowed domains under different hijacked apex domains, using and discarding

them simultaneously, which results in strong but abnormal correlation. However, the legitimate subdomains are more independent from one another. In this study, we measure the correlation from three aspects: first seen date, resolution count, and active days.

Our goal here is to determine how consistent these features are across different subdomains. To this end, we convert each feature into a frequency histogram and compare it to a crafted histogram when all subdomains share the same value, and then use Jeffrey divergence [1] to measure their difference. Specifically, given a set of values V, we first count the weighted frequency of each value, resulting in a set $W = \{< w_i, \frac{w_i}{|V|} >\}_{i=1..n}$. We then derive a new set W' by setting $< w_i, 1 >$ if w_i has the largest frequency $\frac{w_i}{|V|}$; otherwise $< w_i, 0 >$. Finally, Jeffrey divergence is computed over W and W'.

Distribution of first seen date. Given a subdomain s, we compute the Jeffrey divergence of the first seen date (in the format of MM-DD-YYYY) among all subdomains hosted together with s. This feature is denoted as $F11$.

Resolution count. The visits to shadowed domains tend to be more uniform, as they are rotated in regular intervals. The visits to legitimate domains are much more diverse, and certain subdomains like `www` usually receive substantially more visits. Also, legitimate domains tend to receive more visits than malicious ones. To model this property, we define two features, Jeffrey divergence ($F12$) and the reciprocal of median ($F13$) of resolution count.

When computing this feature, we aggregate all the resolution counts associated with the observed IPs for a domain name. Therefore, even if the mapping between an IP address and a domain name is not one-to-one, e.g., when IP-fluxing is played by attackers, the resolution count is not diluted. On the other hand, when an IP is shared across different domain names, e.g., when domain-fluxing is abused, this feature is not affected either, because resolution counts are separated between individual domain names, regardless of their IPs.

Note that while a malicious apex domain is oftentimes mapped to multiple IPs (IP-fluxing), the attackers we studied here usually use subdomains in a thrown-away manner because it costs them nothing to create. More specifically, we observe that a shadowed domain is normally used only for a very short period of time (most of them less than five resolutions) and mapped to one IP.

Active days. The feature above may raise alarm when legitimate subdomains are rarely visited. As a complementary method, we also compute the active days of subdomains, or how long a subdomain and IP pair is witnessed. This works particularly well when an attacker frequently changes the hosting IP. By contrast, IPs for legitimate domains are more stable, resulting in longer active days. Similar to the resolution count, we use two features, Jeffrey divergence ($F14$) and the reciprocal of median ($F15$) of active days.

3.3.4 *Subdomain Name.*

Similar to DGA domains [11, 65, 86], many shadowed domains are algorithmically generated, instead of being manually named. We model the name similarity of all co-hosted subdomains under two numerical features. Note that the randomness of characters (e.g., entropy of words) within one domain name is not considered by us, because we found many shadowed domain do have meaningful label, like `info`.

Diversity of domain name levels. Shadowed domains belonging to the same campaign are usually generated using the same template, and thus their domain levels are the same. However, legitimate domains hosted on the same IP have less uniform domain levels. Similar to the above features, we compute the Jeffrey divergence ($F16$) for all of the subdomains hosted together.

Subdomain name length. For this feature, we remove the substring matching the apex from each subdomain and compare the remaining length. When subdomains in the same group have different levels, we pad them to the maximum level by adding empty strings. Assume the prefix of subdomain is $\mathbb{N} = \{< n_i >_{i=1..m}\}$, where n_i is the i^{th} level, we compute the Jeffrey divergence for each level of name, denoted as Jeffrey($\mathbb{N}_i$), and then take the mean value, as $F17 = \frac{\sum_{i=1}^{m} Jeffrey(N_i)}{m}$.

4 EVALUATION

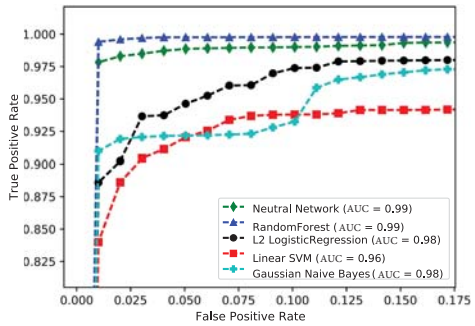
In this section, we present the evaluation results of `Woodpecker` on labeled datasets described in §3.2. We first compare the overall performance of five different classifiers on three ground-truth datasets. Then, we analyze the importance of each feature. Finally, we evaluate `Woodpecker` on two testing sets, $D_{unknown}$ and D_{vt}.

4.1 Training and Testing Classifiers

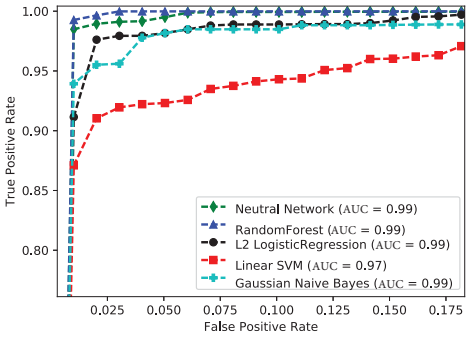
We first test the effectiveness of our detector over the ground-truth datasets, $D_{shadowed}$, D_{pop}, and D_{nonpop} through the standard 10-fold cross-validation. We partition the data based on the apex domains to ensure that for each round of testing, we have subdomains to test from the apex domains unseen in the training phase. Specifically, subdomains in $\frac{9}{10}$ of the randomly selected apex domains fill the training set, and those in the remaining $\frac{1}{10}$ apex domains fill the testing set.

We use the scikit-learn machine-learning library to prototype our classifiers [68]. We compare five mostly used machine-learning classification algorithms, including RandomForest, SVM with a linear kernel, Gaussian Naive Bayes, L2-regularized Logistic Regression, and Neutral Network. Figure 5 illustrates the receiver operating characteristic (ROC) curves of these classifiers, when using Farsight and 360 PDNS to build domain profiles. The x-axis shows the false-positive rate (FPR), which is defined as $\frac{N_{FP}}{N_{FP}+N_{TN}}$, and the y-axis shows the true-positive rate (TPR), which is defined as $\frac{N_{TP}}{N_{TP}+N_{FN}}$. We observe that all classifiers can achieve promising accuracy on both PDNS data sources. To reach a 90% detection rate, the maximum FPR is always less than 3% for all classifiers, suggesting that `Woodpecker` can effectively detect shadowed domains.

Evidently, RandomForest outperforms the other classifiers in all cases. This is mainly because domain shadowing detection is a non-linear classification task. Thus, RandomForest and Neutral Network consistently outperform Logistic Regression and linear SVM. Meanwhile, our dataset is not very clean, e.g., shadowed domains being falsely labeled as benign for training. RandomForest can handle noisy datasets very well [17]. Moreover, some features that `Woodpecker` extracts could be inaccurate, e.g., the resolution count and active days. These features depend on the vantage points



(a) Farsight datasets.



(b) 360 datasets.

Figure 5: Performance comparison of classifiers under 10-fold cross-validation. The number of trees used in RandomForest is 100. All other classifiers use the default configuration in scikit-learn.

where DNS queries are monitored. RandomForest is more robust to those errors [17]. Finally, RandomForest can effectively handle imbalanced training datasets [17].

Next, we draw more details on the false positives and negatives. We focus on the best performing classifier RandomForest only and use it for all follow-up experiments. Due to the space limit, we only present the results on Farsight data (results on 360 have a similar distribution) in the rest of evaluations. In total, `Woodpecker` misclassifies 222 shadowed domains as legitimate (false negatives) and six legitimate ones as shadowed (false positives). We manually inspect these instances to understand the cause of the misclassification. First, about one third of these shadowed domains have snapshots in `Archive.org`. Nevertheless, most of these snapshots were captured several years ago. By contrast, most of the legitimate subdomains in our dataset have much fresher snapshots. For example, the last snapshot of `extranet.melia.com` dated back to 2008, but the subdomain was used for an attack in 2015. We speculate that these subdomains have been abandoned by domain owners (i.e., no longer serving any web content) but were later revived by attackers for illicit purposes. One approach to address this inaccuracy is to set an expiration date for snapshots. Second, the majority of the missed shadowed domains co-host either with siblings only or with a few other subdomains, which lessens the effectiveness of our correlation analysis. On the other hand, the features of all six

Rank	Feature	Score	rank	Feature	Score
1	F10	0.26188	10	F8*	0.03374
2	F2*	0.13213	11	F12*	0.03183
3	F7*	0.11509	12	F16*	0.03128
4	F17	0.06493	13	F3*	0.02852
5	F5*	0.0623	14	F15	0.02395
6	F9	0.05221	15	F13*	0.02309
7	F1*	0.04496	16	F6*	0.01491
8	F14	0.04424	17	F4*	0.00036
9	F11*	0.03451			

Table 5: Importance of features. Features marked with an asterisk (*) are novel.

false positives resemble shadowed domains. For instance, they are all hosted in countries different from their apex domains, and all subdomains on the same IP are visited only a few times.

4.2 Feature Analysis

We assess the importance of our features through a standard metric in the RandomForest model, namely mean decrease impurity (MDI) [18], which is defined as,

$$MDI(X_m) = \frac{1}{N_T} \sum_{T} \sum_{t \in T:(s_t)=X_m} p(t)\Delta f(s_t, t) \quad (3)$$

where X_m is a feature, N_T is the number of trees, p(t) is the proportion of samples reaching node t (N_t/N), $v(s_t)$ is the variable used in split s_t, and $f(s_t, t)$ is an impurity decrease measure (Gini index in our case). Table 5 shows the score of each feature with the novel ones marked with an asterisk. As we can see, three of the top five features are novel, suggesting that using known features is not sufficient to capture shadowed domains.

We further evaluate the impact of different groups of features. Figure 6 compares the performance of `Woodpecker` when deviation-only and correlation-only features are used. Interestingly, `Woodpecker` can still achieve a 95% TPR with less than 0.1% FPR when only features in deviation dimension are used. As such, the operators behind `Woodpecker` can choose to trade a little accuracy for higher efficiency, since computing correlation features are more resource-consuming.

In addition, we assess the performance of features under each of the four categories. The results are shown in Figure 7. Except for the feature of subdomain name, all other feature categories produce a reasonable performance. The feature of subdomain name does not perform well because many legitimate services like cloud platforms and content delivery network (CDN) also have seemingly algorithmically generated domain names.

In summary, according to our analysis, it is almost impossible for attackers to evade `Woodpecker` by manipulating a few features. Instead, they would need to manipulate many features in both deviation and correlation dimensions, and the cost is non-negligible. Take the feature of hosting IP deviation as an example. We observe that most compromised apex domains use their registrars' hosting services. GoDaddy is particularly popular as it is also the largest domain registrar. In order to confuse this feature, attackers can either change the IP of an apex domain, which will be discovered by site owners immediately, or host their shadowed domains on GoDaddy as well. However, unlike less reputable and bullet-proof

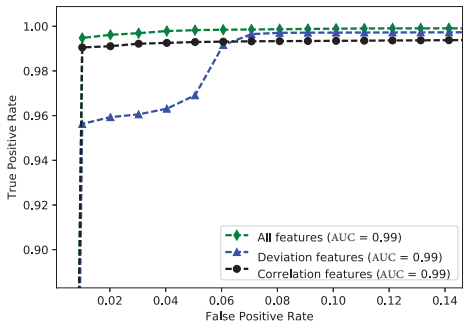


Figure 6: ROC of RandomForest on Farsight data when all, deviation-only (F1, F2, F4, F5, F7) and correlation-only (all others) features are used.

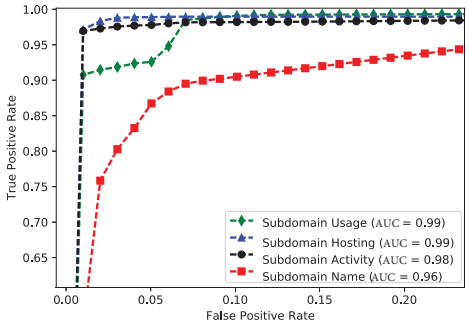


Figure 7: ROC of RandomForest on Farsight data when features in a single category are used.

hosting services, GoDaddy is a poor choice for attackers, due to its much more stringent policies and actions against malicious content.

4.3 Generality of Trained Models

The training and testing stages of our last experiments are carried out on an identical dataset. We want to confirm whether Woodpecker can be trained on one dataset and then applied to another dataset, and how its performance is impacted. To this end, we evaluate two configurations, i.e., training the model on Farsight and testing on 360, and vice versa. We exclude all subdomains in Farsight that overlap with the 360 dataset, and thus the training and testing datasets have no overlap.

Figure 8 illustrates the results when different dimensions of features are used. We find that both configurations cannot produce comparable results to our prior settings when all features are used, which might indicate that Woodpecker needs to be re-trained when being deployed on different vantage points. We further examine the performance when deviation-only features are used. Interestingly, the result of the model trained on Farsight is significantly improved, while the result of the model on 360 remains almost the same. Moreover, the performance of both models decreases significantly when correlation-only features are used. The plausible reason behind this is the uneven coverage of PDNS sources, which greatly impacts the correlation analysis. For instance, given an IP address, Farsight may observe tens of subdomains hosted on the IP, while 360 might observe only one or two. Hence, a model trained on Farsight could derive totally different feature weights compared to 360.

In summary, when deviation-only features are used, Woodpecker can be migrated among different vantage points without re-training. A model trained on a PDNS source would yield better results when tested on the same source.

4.4 Evaluation on $D_{unknown}$

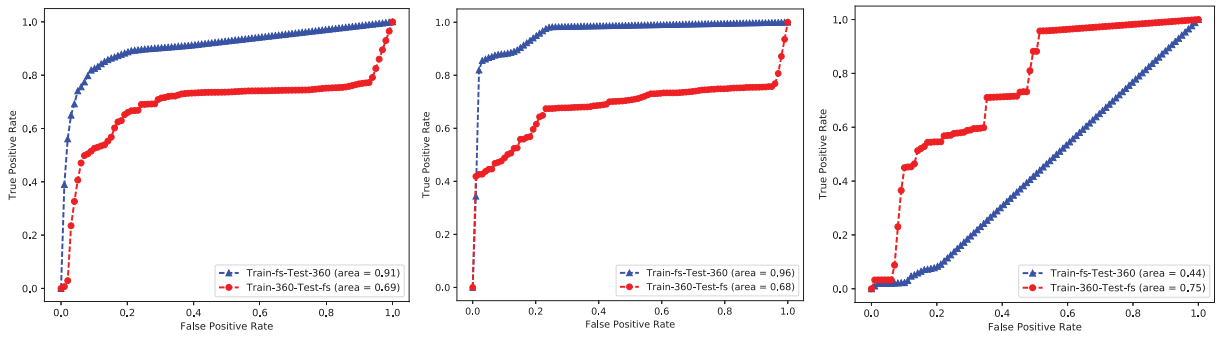
We now evaluate Woodpecker on $D_{unknown}$ to examine whether we can accurately distinguish legitimate and unknown shadowed subdomains under known hijacked apex domains.

Among the 34,586 unknown subdomains in $D_{unknown}$ (Table 1), Woodpecker reports 10,905 shadowed domains. Since this dataset is unlabeled, we have to validate the result through manual investigation. We use a set of rules, after confirming their validity with an analyst from a security company. In particular, we consider a subdomain as a true positive (1) if it has been deleted from the authoritative DNS servers, (2) if it is hosted together with those in $D_{shadowed}$, (3) if its name follows the same pattern as known shadowed ones, (4) if it is reported by other security companies, and (5) if it is not running any legitimate business. After these steps, we confirm 10,866 as true positives and 39 as false detections. The false detection rate is thus 0.35%, which is consistent with our results on $D_{shadowed}$. Measuring FNR is very challenging, given there are still over 20K subdomains remaining. Here we randomly sample 50 apex domains in $D_{shadowed}$ and examine all the subdomains. In the end, we do not find any new shadowed domains missed by Woodpecker.

4.5 Evaluation on D_{vt}

Finally, we apply Woodpecker to a large unlabeled dataset, D_{vt} built from the daily feeds of VT, consisting of more than 20M subdomains that are recorded by 360. This dataset is more representative in that it covers many types of malicious domains, either shadowed or non-shadowed. Many legitimate subdomains are also contained in this dataset. As demonstrated in §4.3, Woodpecker achieves its best performance when it is trained and tested using data from the same PDNS source. Therefore, we use Woodpecker with RandomForest that is trained on 360 data for this evaluation. In total, Woodpecker reports 287,780 shadowed domains (1.28% of the total subdomains) under 23,495 apex domains.

Given these results, we first sanitize them by removing subdomains under malicious apex domains, since our main goal is to detect malicious subdomains created under legitimate apex domains. Then, we verify whether the remaining subdomains are indeed shadowed. Such a validation process is very time-consuming and challenging. The best way is to report all of them to domain owners and registrars and wait for their responses. However, previous studies [52] have shown that most are unresponsive. Even finding all of the recipients is impossible in short term. So, we take a best-effort approach instead and categorize these domains based on clustering and manual analysis. In the end, they can be labeled into five categories.



(a) All features. (b) Deviation-only features. (c) Correlation-only features.

Figure 8: Performance of `Woodpecker` using RandomForest trained and tested on different PDNS sources. FS stands for Farsight.

Expired apex domains. First, we examine the Whois of all apex domains and find that 1,782 out of the 23,495 apex domains have already expired, which account for 45,093 of the reported subdomains. We exclude all subdomains under these expired apex domains, because there is no sufficient information left to us to determine the legitimacy of the apex. This rule may remove some true positives: We check the apex in $D_{shadowed}$ and find that about 18% have expired. As a future improvement, we could run `Woodpecker` more promptly when the data is downloaded from our vantage point.

Lead fraud [48, 55]. Second, we observe that 341 of the *in-use* apex domains covering 86,886 reported subdomains are involved in lead fraud, a type of online scam that solicits user's personal information. They are identified by scanning domain names using a set of keywords attributed to known lead-fraud campaigns, like `rewards`. One such example is `oiyzz.exclusiverewards.6053.ws`. Manual sampling over these domains (and apex) shows most of them are indeed carrying out lead fraud. We check the features of these domains and find that they show similar patterns to domain shadowing. For instance, their subdomains are hosted in different ASes and sometimes in different countries from their apex domains.

Deleted subdomains. After expired and lead fraud domains are excluded, we further run DNS probing over the remaining 155,801 subdomains to see whether they are resolvable. It turns out that 29,565 had already been deleted. We consider these domains very suspicious as their injected DNS records might be purified by domain owners, especially when in most cases their siblings are still resolvable.

Heuristics based pruning. We further validate the remaining resolvable domains using three heuristics. First, we construct the prefix patterns based on known-shadowed domains, which are rarely used by legitimate subdomains, like `add.` and `see.`. Second, we search for the subdomains alarmed by at least one vendor in VT but whose apex domains have no alarms. Third, we cluster all subdomains based on their IP addresses. If one subdomain in a cluster has been confirmed in previous steps, we consider all others to be confirmed as well. In this way, we successfully identify 97,996 additional shadowed domains.

Manual review. Finally, we manually review the remaining 28,240 subdomains. In order to make this task tractable, we cluster these subdomains based on their apex domains and analyze the top 100 large clusters and 200 other random apex domains. We observe that 98 apex domains (covering 14,090 subdomains) are quite suspicious in that we cannot find any information about the hosting sites from Google search results. Meanwhile, many of them have been reported by security companies. Among them, 41 are potential DGA (Domain Generation Algorithm) domains, which we speculate are registered by attackers. In the remaining set, 868 subdomains come from eight dynamic DNS and three CDN services like `dyn-dns.org` and `Limelight CDN`, and they are labeled as false positives. In addition, 358 are falsely alarmed as they run the apex owner's legitimate business, e.g., `live.bilibili.com`, totaling 1,226 false positives. We are unable to confirm the remaining 12,924 subdomains due to their sheer volume.

Summary. In total, 127,561 shadowed domains are confirmed under 21,228 apex domains, hosted on 4,158 IP addresses. Compared to $D_{shadowed}$, only 254 subdomains under 216 apex domains are overlapped. Note that our validation and sanitization of the data is best-effort: True shadowed domains could be eliminated, and legitimate subdomains might be included. We would like to emphasize two lessons learned during this validation process. First, dynamic DNS and CDN services are the main sources of false positives reported by `Woodpecker`. Therefore, to improve accuracy, we have built whitelists for dynamic DNS and CDN services [24, 66]. Second, subdomains under malicious apex domains could exhibit similar features to shadowed domains and trigger alarms. To distinguish them, blacklists focusing on apex domains like VirusTotal and other domain reputation systems [33] can be leveraged. The whitelists and blacklists can be incorporated into `Woodpecker` to further improve its accuracy.

5 MEASUREMENT AND DISCOVERIES

`Woodpecker` identifies in total 127,561 shadowed domains from various sources, which significantly surpasses the community's knowledge about this attack vector (only 26,132 shadowed domains were reported before our study). This sheer amount of data offers us a good opportunity to gain a deeper understanding of this issue. We conduct a comprehensive measurement study on the collected data and report our findings below.

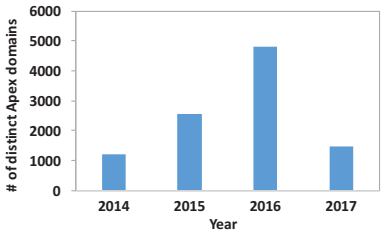


Figure 9: Trend of domain shadowing.

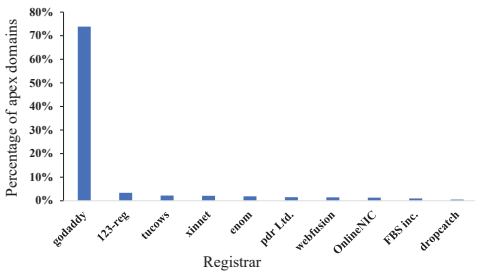


Figure 10: Top 10 registrars in terms of distinct apex domains with shadowed domains.

We first count the number of compromised apex domains, and show the trend in Figure 9. When there are many shadowed domains under an apex domain, we use the year of its first observed shadowed domain. The earliest case that we observed happened in 2014. Since then, the number of affected apex domains increases substantially every year. Because our dataset only contains data before May 2017, we observe fewer shadowed domains in 2017. This result indicates that domain shadowing is becoming increasingly rampant and deserves more attention from the security community. Next, we conduct in-depth analysis from three aspects.

Affected registrars. In total, the shadowed domains trace back to 117 registrars. Figure 10 shows the top 10 registrars in terms of distinct apex domains. We can see that GoDaddy accounts for more than 70% of compromised apex domains while the percentage for other registrars is much lower. Considering that GoDaddy shares about 32% of the domain market, which is much greater than the second largest one (6%), this result does show that domain shadowing is a serious issue for GoDaddy, but this does not necessarily indicate that it is the most vulnerable registrar. There are also small registrars gaining high rankings in our result. The registrant buying domains under them should check their account settings more cautiously.

To assess how these registrars protect their users, we manually examine the security measures of the top 5 registrars. Table 6 shows their password requirements for registrants, whether they enforce two-factor authentication (2FA), and how they notify owners about modifications. We observe that 2FA is either not provided or disabled by default. This situation is alarming and disappointing, as the best account defense does not play a role here. Also, no registrars notify users when the DNS records are modified in the default settings.

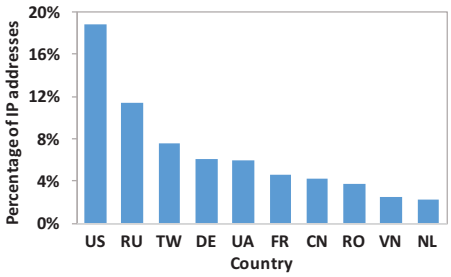


Figure 11: Distribution of IPs in the top 10 countries.

Registrar	Password Length	2FA	Notification of Modifications
GoDaddy	>9 chars with 1 capital, 1 lower and 1 digit	SMS	No
123-reg	>9 chars with 1 capital, 1 digit and 1 special	No	No
Tucows†	-	-	-
XinNet	8-16 with 1 digit	Yes	No
eNom	6-20 with 1 number and 1 special	SMS	No

Table 6: Security policy of the top 5 registrars in our detection. †Tucows is the owner of eNom and Hover etc. and provides services under them.

Hosting IP. In total, 4,158 IP addresses associated with shadowed domains spreading in 91 countries are discovered. Figure 11 illustrates the top 10 countries and their percentages. As shown, most of these IPs are located in the United States (US) and Russia (RU). We further find that the IPs in US and RU are widely spread as they belong to 161 and 137 ASes, respectively. This indicates that domain shadowing is used for many different campaigns or by different attackers. We check these IP addresses in VirusTotal and find that 1,499 IPs were not alarmed. Therefore, malware-evidence or blacklist-based features used in Notos [9] and Kopis [10] will not work well for our settings.

Shadowed domains. Finally, we analyze the characteristics of shadowed domains and their apex. Basically, the number of shadowed domains under an apex is quite random, from one up to 2,989 with the average number at six. Most shadowed domains have a short lifetime and are mostly (85%) resolved for less than five times per IP. Figure 12 shows the CDF of the active days of shadowed domains. Among them, 85% are observed for only one day. This indicates that miscreants rotate shadowed domains quickly, in a similar fashion as fast-flux networks [39].

Previous work [14] uses the TTL value to identify malicious domains. We do not use it for our problem, since it is usually not distinctive on the ground-truth set. We verify this design choice on the entire set by sending DNS queries for 10,000 randomly sampled resolvable shadowed domains. The result confirms our prior observation that the value is either the same as their apex or within the normal range of other legitimate domains.

By cross-checking with VT, we find that 126,384 shadowed domains were submitted to VirusTotal but only 14,134 subdomains were alarmed. In other words, security companies have not yet devised and deployed an effective solution, and we believe that `Woodpecker` can provide great value in tackling domain shadowing.

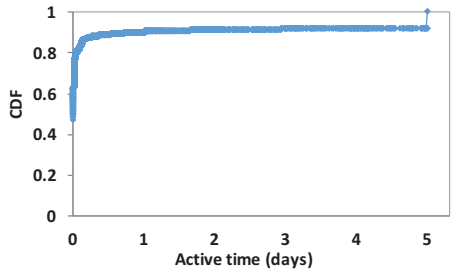


Figure 12: CDF of the active days of shadowed domains.

5.1 Case Studies

There are two new findings uncovered in our measurement study. First, in addition to serving exploit kits, shadowed domains are also used for other attack vectors like phishing. Second, wildcard DNS records are also leveraged to create shadowed domains.

Phishing. All currently reported shadowed domains like those in Table 2 are exclusively involved in exploit kits. However, `Woodpecker` identifies many phishing attempts that exploit shadowed domains. One ongoing campaign is `paypal.com.webapps.`*`random-characters`*`.5degreesfalmouth.co.uk`. We consider the apex domain as legitimate because we find that its Facebook account is actively maintained[7] and it is advertised on reputable websites[8]. There are many similar cases like `verifychase.com-u.mescacompany.com` and `apple.com.`*`random-characters`*`.yclscholarship.org`.

However, we did not see a phishing site impersonating compromised apex domains. We assume this is probably because most compromised apex domains are not popular enough, and only a limited number of victims can be targeted.

Wildcard DNS records. While an arbitrary number of subdomains can be spawned by inserting many `A` and `CNAME` records, the simplest way to create many records is to exploit wildcard DNS records. One prominent advantage of using wildcard records is that attackers do not need to use templates or algorithms to generate subdomain names. However, it is at the cost that the prone to be spotted by domain owners. `Woodpecker` identifies many shadowed domains spawned by wildcard records[9], like `bookstore.hyon.com.cn` and `blackhole.yilaiyin.com`. We determine these cases to be true domain shadowing by incorporating several pieces of evidence. First, most of these apex domains are proven to be legitimate based on the information collected through Google search. Second, all wildcard records under these apex domains point to IP `180.178.59.74`, and several other domains hosted on the IP have at least one alarm in VirusTotal. Our detected subdomains have no alarms because they were never submitted to VirusTotal. Finally, VirusTotal reports that two malware samples communicated with this IP. We observe that all of these apex domains are registered from the same registrar, XinNet. Considering that there have been several data breaches against this registrar [44, 73] in the past, we speculate that these apex domains are probably victims in these incidents.

[7] https://www.facebook.com/5DegreesWest/
[8] https://www.falmouth.co.uk/eatanddrink/5-degrees-west/
[9] Wildcard record is identified if the record `*.apex.com` can be resolved.

6 DISCUSSION

`Woodpecker` is designed to detect subdomains *created in bulk* by attackers. The malicious subdomains falling out of this category might be missed, like modification of existing subdomains or the subdomains created under malicious apex domains, as elaborated in §2.2.

An attacker who knows the features used by `Woodpecker` could change her strategy for evasion. To hinder the effectiveness of our correlation features, the attacker can choose to cut off the connections between the shadowed domains, like spreading them to larger pool of IPs. However, this change would increase the attacker's operational cost. Alternatively, the servers linked to the shadowed domains can be co-hosted with other benign servers on the same set of IPs in order to confuse our detector. So far, we find such co-hosting rarely happened, since many shadowed domains are related to the core components of malicious infrastructures, like exploit servers, which are preferably hosted by bullet-proof providers [32]. In addition, placing the services on reputable hosting providers increases their risk of being captured. To evade our deviation analysis, the attacker can learn how the legitimate services on the apex domains are managed and then configure the shadowed domains to resemble her target. For instance, increasing the observed days until reaching the same level of the apex domain is likely effective against `Woodpecker`. However, such changes are more noticeable to site owners. To summarize, evading `Woodpecker` requires meticulous adjustment from the side of adversary, while the side-effects are inevitable (e.g., raising operational costs and awareness from site owners).

When the subdomains under malicious apex domains exhibit similar features to shadowed domains, they may be detected by `Woodpecker` as well. We believe capturing such instances is also meaningful, especially for security companies. Meanwhile, tools focusing on malicious apex domains, like PREDATOR [33], can be used here for better triaging.

To some extent, the effectiveness of `Woodpecker` depends on the training data. While some previous works rely on data not directly accessible to the public [9, 10, 14], we want to highlight that all of our data is obtained from sources open to researchers and practitioners. Thus, deploying our approach is considerably easier. So far, `Woodpecker` runs in a batch mode, i.e., when PDNS data from a large amount of domains and IPs are available. For real-time detection, `Woodpecker` can be configured to load all existing domain/IP profiles into memory and run the trained model whenever there is an update.

7 RELATED WORK

Detecting malicious domains. A wealth of research has been conducted on detecting malicious domains. Similar to our work, there are different approaches to examining DNS data [9–11, 14, 86]. As elaborated in Appendix §A, shadowed domains exhibit different properties from the objects of previous studies. A new approach is needed, and we show that `Woodpecker` is capable of achieving the detection goal with the combination of deviation and correlation analysis. A recent work by Hao et al. [34] aims to detect malicious domains at their registration time. Given that shadowed domains and their parent apex domains share the same registration

information, such an approach is ineffective at detecting shadowed domains. Plohmann et al. [65] and Lever et al. [50] conducted large-scale studies on malicious domains by running the collected samples in a sandbox environment. Botfinder [76] and Jackstraws [42] aim to detect C&C domains in botnets based on the similar communication patterns of bot clients. By contrast, our approach does not assume the possession of any file samples (malware and web page).

Detecting malicious web content and URLs. Detecting a malicious web page is another active research line in finding traces of cybercriminal activities. Most of the prior works leverage web content and execution traces of a runtime visit for detection. Features regarding web content are deemed effective in detecting web spam [63], phishing sites [85], URL spam [77], and general malicious pages [19]. Malicious sites usually hide themselves behind web redirections, but their redirection pattern is different from legitimate cases, which can be leveraged to spot those malicious sites [49, 74, 84]. Invernizzi et al. [41] showed that a query result returned from search engines can be used to guide the process of finding malicious sites. To trap more visitors, vulnerable sites are frequently compromised and turned into redirectors through code injection. Such a strategy introduces unusual changes to the legitimate sites and can be detected by differing web content [16], HTTP traffic [6], and JS libraries [53]. The URLs associated with malicious web content might exhibit distinctive features, and previous works show machine-learning based approaches are effective at addressing this problem [30, 57, 58]. Obtaining web content or URLs usually requires active web crawling, which is time-consuming and ineffective when cloaking is performed by malicious servers. By contrast, our solution is lightweight and robust against cloaking.

DNS security. Most previous studies on DNS security focus on cache poisoning, which was first uncovered by Bellovin [13] in the 1990s. Conventional cache poisoning attacks exploit the flaws in DNS servers and inject inauthentic RRs to DNS caches. Recently, off-path DNS poisoning has been proposed to poison DNS caches with spoofed DNS responses [35–37, 43]. Alternatively, cybercriminals can set up rogue DNS resolvers so that users' traffic can be arbitrarily rerouted [22, 47]. Domain shadowing is different from cache poisoning and rogue resolvers in that the changes to DNS servers do not exploit their system vulnerabilities. To some extent, domain shadowing resembles the attack that hijacks the dangling DNS records of legitimate domains (called *Dare*) [56]. However, the solution for finding Dare is not viable for our problem, in which there are no dangling DNS records.

Security of domain ecosystem. The security issues in the domain ecosystem, including registrars and registries, have been studied for a long time. In particular, researchers have investigated how domains are recruited and used by attackers for a spectrum of cybercrime businesses, like spam [7], exploit kits [32], blackhat SEO [28], and dedicated hosts [54]. Previous studies also show that adversaries actively register domain names similar to reputable ones (called typosquatting) in hopes of harvesting traffic from careless users [3, 45, 75]. When a domain is not serving its owner's website, the owner could leave it to a parking service that places ads there and share the revenue when the ads are viewed or clicked. However, the business practices of some parking services are problematic, as shown in previous studies [5, 83]. A recent study measures the security on the basis of individual TLD and demonstrates that the scale of free services on a TLD could impact its reputation [46]. Our study is complementary to these existing works in understanding the security issues in the domain ecosystem.

8 CONCLUSION

In this paper, we present the first study on domain shadowing, an emerging strategy adopted by miscreants to build their attack infrastructures. Our study stems from a set of manually confirmed shadowed domains. We find that domain shadowing can be uniquely characterized by analyzing the deviation of subdomains from their apex domains and the correlation among subdomains under different apex domains. Based on these observations, a set of novel features are identified and used to build our domain shadowing detector, `Woodpecker`. Our evaluation on labeled datasets show that among five popular machine-learning algorithms, Random Forest works best, achieving a 98.5% detection rate with an approximately 0.1% false positive rate. By applying `Woodpecker` to the daily feeds of VirusTotal collected in two months, we can detect thousands of new domain shadowing campaigns. Our results are quite alarming and indicate that domain shadowing has become increasingly rampant since 2014. We also reveal for the first time that domain shadowing is not only involved in exploit kits but also in phishing attacks. Another prominent finding is that some miscreants do not use algorithmically generated subdomains but exploit wildcard DNS records.

REFERENCES

[1] 1946. An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences.*

[2] Domain Shadowing With a Twist. 2015. https://blog.malwarebytes.com/threat-analysis/2015/04/domain-shadowing-with-a-twist/.

[3] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven Months' Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS).*

[4] Sumayah Alrwais, Xiaojing Liao, Xianghang Mi, Peng Wang, Xiaofeng Wang, Feng Qian, Raheem Beyah, and Damon McCoy. 2017. Under the Shadow of Sunshine: Understanding and Detecting BulletProof Hosting on Legitimate Service Provider Networks. In *IEEE S&P.*

[5] Sumayah Alrwais, Kan Yuan, Eihal Alowaisheq, Zhou Li, and XiaoFeng Wang. 2014. Understanding the Dark Side of Domain Parking. In *USENIX Security Symposium (USENIX Security).*

[6] Sumayah Alrwais, Kan Yuan, Eihal Alowaisheq, Xiaojing Liao, Alina Oprea, XiaoFeng Wang, and Zhou Li. 2016. Catching Predators at Watering Holes: Finding and Understanding Strategically Compromised Websites. In *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC).*

[7] David S. Anderson, Chris Fleizach, Stefan Savage, and Geoffrey M. Voelker. 2007. Spamscatter: Characterizing Internet Scam Hosting Infrastructure. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium (SS'07).*

[8] Fake Extensions Angler EK: More Obfuscation and Other Nonsense. 2015. http://blogs.cisco.com/security/talos/angler-update.

[9] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. 2010. Building a Dynamic Reputation System for DNS. In *Proceedings of the 19th USENIX Conference on Security.*

[10] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, II, and David Dagon. 2011. Detecting Malware Domains at the Upper DNS Hierarchy. In *Proceedings of the 20th USENIX Conference on Security.*

[11] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. 2012. From Throw-away Traffic to Bots: Detecting the Rise of DGA-based Malware. In *Proceedings of the 21st USENIX Conference on Security Symposium.*

[12] Internet Archive. 2017. https://archive.org/.

[13] Steven M. Bellovin. 1995. Using the Domain Name System for System Break-ins. In *USENIX Security.*

[14] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)*.

[15] Website blocked as malicious. 2015. https://forum.avast.com/index.php?topic=167705.0/.

[16] Kevin Borgolte, Christopher Kruegel, and Giovanni Vigna. 2013. Delta: Automatic Identification of Unknown Web-based Infection Campaigns. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS)*.

[17] Leo Breiman and Adele Cutler. 2017. Random Forests. In *https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm*.

[18] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees*. CRC press.

[19] Davide Canali, Marco Cova, Giovanni Vigna, and Christopher Kruegel. 2011. Prophiler: A Fast Filter for the Large-scale Detection of Malicious Web Pages. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*.

[20] Sundown EK: You Better Take Care. 2016. http://blog.talosintelligence.com/2016/10/sundown-ek.html.

[21] CommonCrawl. 2017. http://commoncrawl.org/.

[22] David Dagon, Chris Lee, Wenke Lee, and Niels Provos. 2008. Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority. In *NDSS*.

[23] defintel. 2016. Shadow Puppets - Domain Shadowing 101. https://defintel.com/blog/index.php/2016/03/shadow-puppets-domain-shadowing-101.html. (2016).

[24] Dynamic DNS. 2017. https://doc.pfsense.org/index.php/Dynamic_DNS.

[25] Forward DNS. 2017. https://scans.io/study/sonar.fdns_v2.

[26] DNSDB. 2017. https://www.farsightsecurity.com/solutions/dnsdb/.

[27] Peru domain registrar hacked & 207116 domain credentials stolen. 2012. https://www.alertlogic.com/blog/peru-domain-registrar-hacked-and-207,116-domain-credentials-stolen-anonymous-group/.

[28] Kun Du, Hao Yang, Zhou Li, Haixin Duan, and Kehuan Zhang. 2016. The Ever-Changing Labyrinth: A Large-Scale Analysis of Wildcard DNS Powered Blackhat SEO. In *USENIX Security Symposium (USENIX Security)*.

[29] David Dunkel. 2015. Catch Me If You Can: How APT Actors Are Moving Through Your Environment Unnoticed. http://blog.trendmicro.com/catch-me-if-you-can-how-apt-actors-are-moving-through-your-environment-unnoticed/. (2015).

[30] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. 2010. On the Potential of Proactive Domain Blacklisting. In *Proceedings of the USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET)*.

[31] Security Alert: Angler EK Accounts for Over 80% of Drive-by Attacks in the Past Month. 2016. https://heimdalsecurity.com/blog/angler-exploit-kit-over-80-of-drive-by-attacks/.

[32] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, M. Zubair Rafique, Moheeb Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. 2012. Manufacturing Compromise: The Emergence of Exploit-as-a-service. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS '12)*.

[33] Shuang Hao, Alex Kantchelian, Brad Miller, Vern Paxson, and Nick Feamster. 2016. PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[34] Shuang Hao, Matthew Thomas, Vern Paxson, Nick Feamster, Christian Kreibich, Chris Grier, and Scott Hollenbeck. 2013. Understanding the Domain Registration Behavior of Spammers. In *ACM IMC*.

[35] Amir Herzberg and Haya Shulman. 2012. Security of Patched DNS. In *ESORICS*.

[36] Amir Herzberg and Haya Shulman. 2013. Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org. In *IEEE CNS*.

[37] Amir Herzberg and Haya Shulman. 2013. Socket Overloading for Fun and Cache-poisoning. In *ACSAC*.

[38] Tobias Holgers, David E. Watson, and Steven D. Gribble. 2006. Cutting Through the Confusion: A Measurement Study of Homograph Attacks. In *USENIX ATC*.

[39] Thorsten Holz, Christian Gorecki, Konrad Rieck, and Felix C. Freiling. 2008. Measuring and Detecting Fast-Flux Service Networks. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)*.

[40] Threat Spotlight: Angler Lurking in the Domain Shadows. 2015. http://blogs.cisco.com/security/talos/angler-domain-shadowing.

[41] Luca Invernizzi, Stefano Benvenuti, Marco Cova, Paolo Milani Comparetti, Christopher Kruegel, and Giovanni Vigna. 2012. EvilSeed: A Guided Approach to Finding Malicious Web Pages. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

[42] Gregoire Jacob, Ralf Hund, Christopher Kruegel, and Thorsten Holz. 2011. JACKSTRAWS: Picking Command and Control Connections from Bot Traffic. In *Proc. 20th USENIX Security Symposium*.

[43] D. Kaminsky. 2008. It's the End of the Cache As We Know It. In *Blackhat Briefings*.

[44] Kankanews. 2014. Xinnet breach leads false resolution of registered sites. http://www.kankanews.com/a/2014-04-02/0014513245.shtml. (2014).

[45] Mohammad Taha Khan, Xiang Huo, Zhou Li, and Chris Kanich. 2015. Every Second Counts: Quantifying the Negative Externalities of Cybercrime via Typosquatting. In *IEEE Symposium on Security and Privacy (S&P)*.

[46] Maciej Korczynski, Samaneh Tajalizadehkhoob, Arman Noroozian, Maarten Wullink, Cristian Hesselman, and Michel van Eeten. [n. d.]. Reputation Metrics Design to Improve Intermediary Incentives for Security of TLDs. In *Proceedings of 2nd IEEE European Symposium on Security and Privacy (Euro S&P)*.

[47] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. 2015. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *ACM IMC*.

[48] How lead fraud happens? 2015. https://www.databowl.com/blog/posts/2015/10/07/how-lead-fraud-happens.html.

[49] Nektarios Leontiadis, Tyler Moore, and Nicolas Christin. 2011. Measuring and Analyzing Search-redirection Attacks in the Illicit Online Prescription Drug Trade. In *Proceedings of USENIX Conference on Security*.

[50] Chaz Lever, Platon Kotzias, Davide Balzarotti, Juan Caballero, and Manos Antonakakisz. 2017. A Lustrum of Malware Network Communication: Evolution and Insights. In *38th IEEE Symposium on Security and Privacy (S&P)*.

[51] Chaz Lever, Robert Walls, Yacin Nadji, David Dagon, Patrick McDaniel, and Manos Antonakakis. 2016. Domain-Z: 28 Registrations Later Measuring the Exploitation of Residual Trust in Domains. In *IEEE Symposium on Security and Privacy (SP)*.

[52] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon McCoy, Stefan Savage, and Vern Paxson. 2016. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *USENIX Security Symposium*.

[53] Zhou Li, Sumayah Alrwais, Xiaofeng Wang, and Eihal Alowaisheq. 2014. Hunting the Red Fox Online: Understanding and Detection of Mass Redirect-Script Injections. In *IEEE Symposium on Security and Privacy (S&P)*.

[54] Zhou Li, Sumayah Alrwais, Yinlian Xie, Fang Yu, and Xiaofeng Wang. 2013. Finding the Linchpins of the Dark Web: a Study on Topologically Dedicated Hosts on Malicious Web Infrastructures. In *IEEE Symposium on Security and Privacy (S&P)*.

[55] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2012. Knowing Your Enemy: Understanding and Detecting Malicious Web Advertising. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS)*.

[56] Daiping Liu, Shuai Hao, and Haining Wang. 2016. All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[57] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. 2009. 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)*.

[58] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. 2009. Identifying Suspicious URLs: An Application of Large-scale Online Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*.

[59] Let's Encrypt Now Being Abused By Malvertisers. 2016. http://blog.trendmicro.com/trendlabs-security-intelligence/lets-encrypt-now-being-abused-by-malvertisers.

[60] Malware-Traffic-Analysis. 2017. 2017-04-06 - EITEST RIG EK from 109.234.36.165 sends matrix ransomware variant. http://www.malware-traffic-analysis.net/2017/04/06/index2.html. (2017).

[61] Alexa Top 1 Million. 2017. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.

[62] Mozilla. 2017. Public suffix list. https://publicsuffix.org/list/public_suffix_list.dat. (2017).

[63] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting Spam Web Pages Through Content Analysis. In *Proceedings of the 15th International Conference on World Wide Web (WWW)*.

[64] PassiveDNS. 2017. http://netlab.360.com/.

[65] Daniel Plohmann, Khaled Yakdan, Michael Klatt, Johannes Bader, and Elmar Gerhards-Padilla. 2016. A Comprehensive Measurement Study of Domain Generating Malware. In *25th USENIX Security Symposium*.

[66] CDN IP ranges. 2017. https://zenodo.org/record/842988#.WZJtrVGGMzM.

[67] Domain registrar attacked customer passwords reset. 2013. http://www.theregister.co.uk/2013/05/09/name_dot_com_data_leak/.

[68] scikit learn. 2017. http://scikit-learn.org/.

[69] The shadow knows: Malvertising campaigns use domain shadowing to pull in Angler EK. 2015. https://www.proofpoint.com/us/threat-insight/post/The-Shadow-Knows/.

[70] Malvertising slowing down but not out. 2016. https://blog.malwarebytes.com/cybercrime/exploits/2016/07/malvertising-slowing-down-but-not-out/.

[71] Threat spotlight: CISCO TALOS thwarts access to massive international exploit kit generating $60M annually from ransomware alone. 2015. http://www.talosintelligence.com/angler-exposed/.

[72] Tom Spring. 2016. Inside the RIG exploit kit. https://threatpost.com/inside-the-rig-exploit-kit/121805/. (2016).

[73] The story around the Linode hack. 2013. https://news.ycombinator.com/item?id=5667027.
[74] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS)*.
[75] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. 2014. The Long "Taile" of Typosquatting Domain Names. In *USENIX Security Symposium (USENIX Security)*.
[76] Florian Tegeler, Xiaoming Fu, Giovanni Vigna, and Christopher Kruegel. 2012. BotFinder: Finding Bots in Network Traffic Without Deep Packet Inspection. In *Proc. 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '12)*.
[77] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. 2011. Design and Evaluation of a Real-Time URL Spam Filtering Service. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.
[78] Talos ShadowGate Take Down: Global Malvertising Campaign Thwarted. 2016. http://blog.talosintelligence.com/2016/09/shadowgate-takedown.html.
[79] Hover Resets User Passwords Due to Possible Breach. 2015. http://www.securityweek.com/hover-resets-user-passwords-due-possible-breach/.
[80] Angler Attempts to Slip the Hook. 2016. http://blog.talosintelligence.com/2016/03/angler-slips-hook.html.
[81] A Look Into Malvertising Attacks Targeting The UK. 2016. https://blog.malwarebytes.com/threat-analysis/2016/03/a-look-into-malvertising-attacks-targeting-the-uk/.
[82] VirusTotal. 2017. https://www.virustotal.com/.
[83] Thomas Vissers, Wouter Joosen, and Nick Nikiforakis. 2015. Parking Sensors: Analyzing and Detecting Parked Domains. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)*.
[84] David Y. Wang, Stefan Savage, and Geoffrey M. Voelker. 2011. Cloak and Dagger: Dynamics of Web Search Cloaking. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS)*.
[85] Colin Whittaker, Brian Ryner, and Marria Nazif. 2010. Large-Scale Automatic Classification of Phishing Pages. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)*.
[86] Sandeep Yadav, Ashwath Kumar Krishna Reddy, A.L. Narasimha Reddy, and Supranamaya Ranjan. 2010. Detecting Algorithmically Generated Malicious Domain Names. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC)*.

A EXISTING SYSTEMS

- Antonakakis et al. [9] proposed a system named `Notos` to dynamically assign reputation scores for domain names. `Notos` uses three categories of features to check a domain `d`, namely network-based (i.e., IPs associated with `d`), zone-based (i.e., subdomains under `d`), and evidence-based (i.e., malware samples contacting `d`). Zone-based features are useful in measuring the apex domain but not for individual subdomains. Most of the shadowed domains in our ground-truth dataset and testing dataset are related to drive-by-download and phishing activities, which are not directly contacted by malware. Hence, evidence-based features are ineffective here.

- `Exposure`, a system developed by Bilge et al. [14] shares the same goal as `Notos`. But different from `Notos`, it does not require any historical data associated with malicious activities and is able to detect malicious domains from an unseen IP. The key insight is that malicious domains exhibit different statistical properties aggregated among requests, e.g., the repeated querying pattern, the diversity of associated IPs, and the low average TTL. However, we found that many shadowed domains do not share the same properties; they are thrown away quickly after going live for a short window, pointed to one IP during its lifetime and bounded to a regular TTL.

- `Kopis` was developed by Antonakakis et al. [10] to detect malicious domains using DNS traffic logged by a single upper-level DNS server, like a TLD server or an authoritative name server. Different from `Notos` and `Exposure`, `Kopis` requires very fine-grained DNS data, like the timestamp and source IP of a single domain request, instead of aggregated data. We argue that such data provides higher visibility but is hardly accessible to parties other than DNS operators. So far, we have not found any public sharing programs of DNS logs from well-known DNS operators. Other issues with `Kopis` include its dependence on evidence (not available for shadowed domains) and its prerequisite of diversity of requesters (many shadowed domains are visited only a few times as observed in our data).

- `Pleiades` detects domains used by DGA-based botnets, based on the insight that bot clients tend to query a large amount of domains, but only a few of them actually resolve to IP addresses (while others return `NXDOMAIN` responses) [11]. This observation does not hold in our case, where most of the subdomains we found were resolvable at some point.

- Yadav et al. proposed a system to detect DGA domains by computing the distribution of alphanumeric characters of domains in an IP-domain cluster [86]. Since algorithmically generated domains present different distributions compared to domains created for legitimate purposes, they can be effectively detected. However, an adversary in our case can use any names for the labels under the apex domain level as long as these names are not used by the domain owner. The name can be short but meaningful, like `info`, which becomes a blind spot for the DGA detector.

In summary, shadowed domains exhibit different features (e.g., ephemeral and readable names) and are used for many attack vectors (e.g., exploit kit and phishing, instead of only botnet-related attacks). Thus, the problem of domain shadowing cannot be addressed by the existing systems.