Compromised or Attacker-Owned: A Large Scale Classification and Study of Hosting Domains of Malicious URLs

Ravindu De Silva[†‡], *Mohamed Nabeel*[‡], *Charith Elvitigala*[†], *Issa Khalil*[‡],
Ting Yu[‡], *Chamath Keppitiyagama*[*]
[†]*SCoRe Lab*
[‡]*Qatar Computing Research Institute*
[*]*University of Colombo School of Computing*
ravindud@scorelab.org, mnabeel@hbku.edu.qa, charitha@scorelab.org, {ikhalil,tyu}@hbku.edu.qa, chamath@ucsc.cmb.ac.lk

Abstract

The mitigation action against a malicious website may differ greatly depending on *how* that site is hosted. If it is hosted under a private apex domain, where all its subdomains and pages are under the apex domain owner's direct control, we could block at the apex domain level. If it is hosted under a public apex domain though (e.g., a web hosting service provider), it would be more appropriate to block at the subdomain level. Further, for the former case, the private apex domain may be legitimate but compromised, or may be attacker-generated, which, again, would warrant different mitigation actions: attacker-owned apex domains could be blocked permanently, while only temporarily for compromised ones.

In this paper, we study over eight hundred million VirusTotal (VT) URL scans from Aug. 1, 2019 to Nov. 18, 2019 and build the first content agnostic machine learning models to distinguish between the above mentioned different types of apex domains hosting malicious websites. Specifically, we first build a highly accurate model to distinguish between public and private apex domains. Then we build additional models to further distinguish compromised domains from attacker-owned ones. Utilizing our trained models, we conduct a large-scale study of the host domains of malicious websites . We observe that even though public apex domains are less than 1% of the apexes hosting malicious websites, they amount to a whopping 46.5% malicious web pages seen in VT URL feeds during our study period. 19.5% of these public malicious websites are compromised. Out of the remaining websites (53.5%), which are hosted on private apexes, we observe that attackers mostly compromise benign websites (65.6%) to launch their attacks, whereas only 34.4% of malicious websites are hosted on domains registered by attackers. Overall, we observe the concerning trend that the majority (81.7%) of malicious websites are hosted under apex domains that attackers do not own.

1 Introduction

Every week millions of users are tricked into access malicious websites from where miscreants launch various attacks including phishing, spams, and malware [14, 19]. Even with recent advances in techniques and tools to detect malicious websites [1, 20, 35, 70], many malicious websites are undetected or detected much later after the damage is done [4]. One key reason for this negative trend is that, instead of registering their own domains, attackers are increasingly hosting their websites on infrastructures they do not own, evading detection by current reputation systems [42]. While the detection of malicious websites, especially phishing and malware websites registered by attackers, have been extensively studied [23, 28, 35], very little has been done to analyze *how* these malicious websites are hosted. Knowing this early greatly helps security professionals take appropriate mitigation actions. Specifically, this paper is motivated by the following questions:

- To what extent attackers host their websites in what we call *public* apex domains such as free web hosting, document sharing or dynamic DNS services? Are they *attacker-owned* or *compromised*?
- For the remaining malicious websites, are they hosted on *compromised* domains or *attacker-owned* domains? To what extent?
- Do the above four hosting types have different characteristics in terms of attack types, duration, volume, and reputation?
- How can we proactively detect these different hosting types of malicious websites?

We first make an important distinction between *public* and *private* apex domains. A public apex domain hosts websites that are not created by and not under the direct control of the apex domain owner, whereas a private apex domain always hosts websites under the control of the domain owner. For

example, 000webhostapp.com is a public apex domain, and alice.000webhostapp.com is a subdomain whose content is not controlled by 000webhostapp.com owner, but by an entity Alice that uses 000webhostapp.com's service. While the majority of public websites are created utilizing prefixes like above, some public websites are created with path suffixes (e.g., sites.google.com/site/alice). nsa.gov, on the other hand, is a private apex domain, and careers.nsa.gov is a subdomain that is clearly under the control of the NSA. The distinction between public and private apex domains has a profound impact on the inference and prediction of malicious domains, especially when it relies on the association of subdomains belonging to the same apex domain [29, 64]. Further, once malicious websites are detected, the actions against the hosting apex domains would be different depending on whether they are public or private.

Though there exist lists of public apex domains from multiple sources, they are by no means complete. Even combined, they account for less than 20% of the public apex domains that our study identifies. Further, these lists are often not up to date due to the highly dynamic nature of the public web-hosting and cloud business. Thus, given a malicious URL, we could not simply look up such lists to decide whether it is hosted in public apex domains. In this work, we design a machine learning model to accurately classify whether a malicious website is hosted in a public or private apex domain. Our key observation is that subdomains of private apex domains have more consistent behavior and properties compared to those of public apex domains.

Once a malicious website is identified as hosted in a public apex domain, we classify the public website based on its owner as either attacker owned (e.g. fbook-png.000webapphost.com and sites.google.com/site/bitcoin2me2) or compromised (e.g. 2014-healthyfood.blogspot.com and sites.google.com/site/kailyali). Similarly, for each website hosted on a private apex, we further classify the apex domain based on its owner. A malicious website is either created by attackers on their own registered domains (e.g., getbinance.org) or on compromised benign domains (e.g., questionpro.com). In the latter case, legitimate domains exploited for malicious activities are victim domains. Takedown strategies and who should be contacted differ depending on the type of the apex domain. Detection of compromised domains early helps owners identify the root causes of the security breach, take corrective measures, and control reputation damage, while SOC (Security Operation Center) teams may temporarily block such victim domains to protect their users. On the other hand, attacker-owned domains would require completely different actions. They are usually first blacklisted to contain the immediate damage. They could be further shut down through third-party takedown services [8, 12], domain registration deletion [7], or ownership transferring if they are involved in cybersquatting [2].

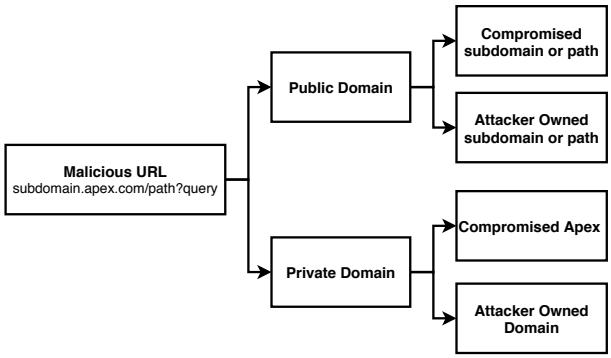


Figure 1: Classification of malicious URLs

Most research in malicious domains focuses on characterizing or detecting attacker owned domains [26, 34, 35, 43, 61, 66]. There have been only a few efforts to either classify compromised phishing domains [30] or to make a distinction between compromised and attacker created phishing or malware domains [45, 52]. Most of these approaches first filter a limited number of public domains based on known public domain lists, and then rely on the contents of websites to build models using data that are often difficult to collect, e.g., multiple snapshots of a website from the Internet Wayback Machine [13, 52] or all the pages belonging to a website [30]. A notable effort on detecting compromised domains is recently introduced by Liu et. al [42] where they build a system called Woodpecker to train a classifier based on passive DNS data and web connectivity graphs to identify compromised subdomains on private apexes, which they term shadow domains. Their goal is different from ours as they profile the behavior of benign domains and then identify those deviating from the profile as shadow domains. In contrast, our work's key goal is to accurately identify malicious domains as either compromised, including shadow and path suffix-based websites or attacker-owned. Nevertheless, building on top of the knowledge gained from these prior work, in this paper we design the first machine learning classifiers to accurately differentiate malicious websites hosted on compromised domains from those on attacker-owned domains for both public and private apexes.

In summary, as shown in Figure 1, our work automatically labels malicious websites (i.e., URLs) as hosted on either public or private apexes. For public websites, we identify attacker-owned subdomains/path prefixes from compromised ones. For private websites, we label them as compromised or attacker-owned apexes.

In this work, we utilize URLs that appeared in VirusTotal (VT) URL feed from Aug. 1, 2019 to Nov. 18, 2019 as our main dataset. VT [67] is a state-of-the-art reputation service that provides aggregated intelligence on any URL by consulting over 70 third-party anti-virus tools and URL/domain reputation services. We refer to each of these tools as a scanner. VT aggregates the query results every second and makes

them available for subscribed users as a feed. Thus, our dataset contains all URL queries submitted to VT worldwide during the above mentioned time period. A basic measure of maliciousness from VT results is the number of scanners that mark a URL as malicious. The higher this value is for a given URL, the more likely the URL is malicious. Based on prior research [59, 68] and our empirical analysis, we consider any URL marked by 5 or more scanners as malicious. Note that, though we use VT as the main source of intelligence of malicious domains, our approach is general and can be easily adapted to work with other malicious domain intelligence sources, as will be discussed in section 6.1.

Specifically, we make the following three broad contributions:

A new classification of public and private apex domains. Whether a website is hosted in a public or private apex domain has an important implication in security practice. We design the first classifier to classify public and private apex domains utilizing historical VT URL feed information. Our classifier achieves 97.2% accuracy with 97.7% precision and 95.6% recall.

New classification schemes to differentiate compromised and attacker-owned domains appearing in VT. When scanners mark a website (hosted in either public or private apexes) as malicious, it is not apparent if its hosting domain is in fact compromised or attacker-owned. We take the first steps to automatically make this distinction with high accuracy. For the classification of malicious private websites, our classifier achieves 96.4% accuracy with 99.1% precision and 92.6% recall. Our classifier for public malicious websites achieves 97.1% accuracy with 97.2% precision and 98.1% recall.

A detailed analysis of public/private apex domains and compromised/attacker-owned domains in VT URL Feed. Based on our trained machine learning models, we analyze the detected public/private apex domains and compromised/attacker-owned domains to gain insights on the malicious websites seen on VT URL feed, which we believe help steer future research on the detection of malicious websites.

The rest of the paper is organized as follows. Section 2 provides information on data sources and preliminaries. Section 3 gives an overview of the overall approach proposed in our work. In Section 4, we provide detailed information about our data source, VT URL feed, and characterize its behavior over time. Section 5 contains the crust of our work, where we detail the classifiers we build and their performances. In Section 6, we then analyze the classifiers under various aspects including robustness, concept drift, and the quality and quantity of the training data. Section 7 discusses the lessons learned and the limitations of our work. Finally, in Section 9, we conclude the paper.

2 Data Sources and Preliminaries

2.1 Public and Private Apex Domains

As mentioned before, we categorize e2LD (effective Second Level Domain) domains as public and private apex domains. An apex domain is public if its subdomains or path suffixes are not created and not under the control of the apex domain owner. Similarly, an apex domain is private if its subdomains are created and managed by the apex domain owner. Accurately identifying these two types of apex domains help SOC teams to take appropriate measures if they are found to be malicious.

2.2 VT URL Feed and Scanners

VT provides one of the most popular URL scanning services widely used in both academia and industry [71]. VT's URL scanning service simply pushes a querying URL to over 70 third-party scanners and gives the aggregated results back. A basic measure of a VT report is the number of scanners that mark a given website as malicious. Also, each scanner labels a malicious URL with one of the following attack types: malicious, phishing, mining, malware, or suspicious. In this study, we consider any URL marked by 5 or more scanners as malicious.

We have built a system called VT NOD/NOH (Newly Observed Domains/Hosts) to profile domains observed in VT URL feed continuously. NOD and NOH incrementally build an aggregated record for each apex and FQDN (Fully Qualified Domain Name). The record includes the time first seen, the time last seen (the timestamp the apex/FQDN is first and last scanned in VT), the number of times scanned, the number of times marked malicious, corresponding URLs, and VT scan summaries. We use VT NOD/NOH to extract features to build our machine learning models described in Section 5.

2.3 Passive DNS Data Feed

Passive DNS (PDNS) [69] captures traffic by sensors cooperatively deployed in various DNS hierarchy locations. For example, Farsight PDNS data [32] utilizes sensors deployed behind DNS resolvers and provides aggregate information about domain resolutions. In our research, we use Farsight PDSN DB to extract PDNS related features for our classifiers.

Among other information, the PDNS DB contains a set of summarized records for each FQDN. Each summarized record contains the time of first seen and last seen (i.e., timestamps of the first and the latest resolution of an FQDN), the number of times the FQDN is queried, resolved IP addresses, and the authoritative name servers. We can extract important hosting features from the PDNS DB, as described in Section 5, to train our classifiers.

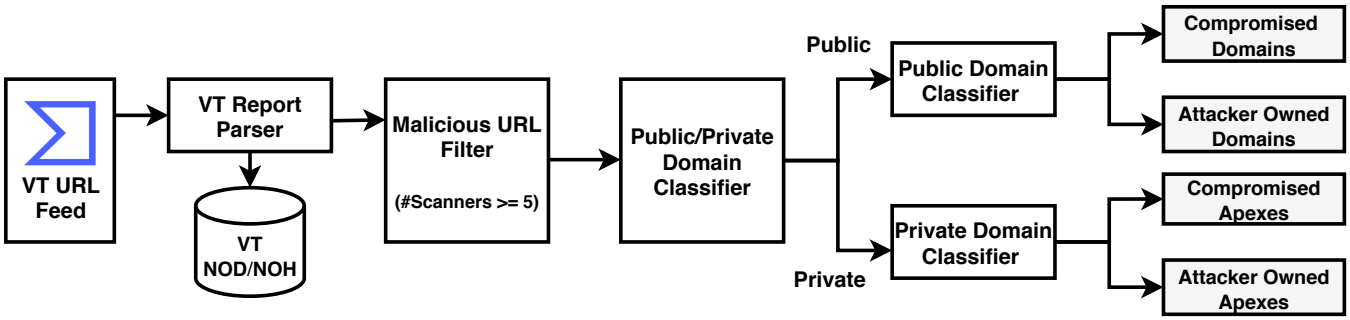


Figure 2: Overall Workflow of Labeling Malicious Websites

2.4 Other Blacklists/Scanners

In addition to VT, we further utilize four major blacklists and reputation systems: Google Safe Browsing (GSB) [16], Phishtank [51], Anti-Phishing Working Group (APWG) [3], and McAfee Site Advisor (SA) [18]. While Phishtank only focuses on phishing websites, the other three systems provide a reputation on any type of malicious websites. Phishtank and APWG maintain a list of manually verified phishing websites. We utilize these websites' results to manually label our dataset as most of these blacklists provide additional textual information about the details of the malicious activities on a website.

2.5 Naive Approaches

After identifying and filtering public domains, one of our work's primary goals is to categorize malicious websites as hosted on compromised or attacker-owned apex domains. A seemingly compelling approach is to take domain popularity, such as Alexa ranking [22] into consideration. It is generally understood that compromised domains have some residual reputation and are long-lived, whereas attacker-owned domains have a low reputation and are short-lived. However, our analysis of the malicious websites in VT shows that such observations do not always hold. While there are compromised domains that have high Alexa ranking and long lifetime (e.g., `linode.com`, `cleverreach.com`), a worrying fact we observe is that there exist many other likely abandoned or little maintained domains with low or no Alexa ranking (e.g., `gemtown88.com`, `vanemery.com`) that are compromised by attackers to launch their attacks. Further, newly created benign domains possess neither of the above properties, making them likely mislabeled as attacker-owned when they are, in fact, compromised. On the other hand, though it is certainly true that many domains created by attackers are short-lived with very low Alexa rankings, sophisticated attackers nowadays increasingly utilize long-lived domains, for example, by creating and parking those domains for a while (e.g., `crackarea.com`, `estilo.com.ec`) to evade detection. Additionally, attackers can artificially inflate the popularity of their domains, at least in the short term, without requiring much investment [58]. Therefore, relying on the popularity and/or lifetime alone does not accurately classify compromised and attacker-owned domains.

One may wonder whether VT reports contain sufficient information to classify the types of hosting domains. We analyze the features built from VT reports, and observe that only with those features a classifier could not achieve sufficient accuracy.

Table 1: VT URL stats for the two datasets

Dataset	#scanners = 0	#scanners $\geq$ 1	#scanners $\geq$ 5
DS1	47,182,496	7,330,850	3,434,226
DS2	37,323,778	9,797,649	4,398,584

3 Overview

Figure 2 illustrates our overall workflow of labeling malicious websites as hosted on public or private apexes, then as compromised or attacker-owned for each category. We explain each step in the workflow in detail.

3.1 VT URL Feed

VT URL scans issued from all over the world are aggregated into hourly feed files. Our system pulls these hourly files, parse them and build profiles for each apex/FQDN observed over time (VT NOD/NOH). While we primarily utilize VT URL feed as the input data source, one may utilize other blacklists as the starting point as demonstrated in Section 6.1. On average, there are 4.8M unique URLs each day in the VT URL feed, out of which a vast majority (89.7%) are likely benign (#scanners = 0, i.e., none of the scanners mark them as malicious). We select two different datasets, DS1 (Aug. 01, 2019 to Aug. 19, 2019) and DS2 (Oct 01 2019 to Oct 14 2019) that are temporally disjoint and of different window sizes to train machine learning models on different datasets and show their generalizability. Table 1 summarizes the VT URL statistics of the two datasets.

Table 2: Malicious domain stats for the two datasets

Dataset	Malicious URLs		Malicious Apexes	
	#public	#private	#public	#private
DS1	1,669,033	1,765,192	3,480	369,758
DS2	2,137,711	2,260,872	3,195	355,567

3.2 Malicious URLs Filter

Out of all URLs marked by at least one scanner (i.e., #scanners ≥ 1), we identify a subset of URLs that are highly likely to be malicious for this study. In order to decide what threshold of #scanners should be used to deem a URL malicious, we take a random sample of 500 of these VT URLs and manually check if they are malicious. Based on this experiment, we identify that VT URLs with 5 or scanners assessing them as malicious are highly likely to be malicious, which is in fact reinforced by prior research findings [59, 68]. Thus, we set the #scanners to 5 or more to extract malicious URLs for the next stage of the pipeline.

3.3 Public/Private Domain Classifier

Malicious URLs identified in the previous step are fed to our public/private domain classifier, which we describe in detail in Section 5.1. This classifier identifies and labels URLs hosted on public and private apex domains with high accuracy. Table 2 shows in each of the two datasets the number of malicious (i.e. those with #scanners ≥ 5) public and private URLs and the number of unique public and private apex domains hosting these URLs. Notice that though the number of unique public apex domains are low, the number of malicious URLs they host is close to that of those hosted by private apex domains, as each public domain hosts a huge number of malicious URLs.

3.4 Private Apex Classifier

In one of the two final stages of the pipeline, we label the identified private apex domains as either compromised or attacker-owned. We train a machine learning model utilizing features from several disparate sources, detailed in Section 5.2. We achieve an accuracy of 96.4% with 99.1% precision and 92.6% recall. We extract the features for each URL under consideration and feed them to the trained machined learning model to predict its label.

3.5 Public Domain Classifier

In this final stage, we label identified public domains as either compromised or attacker-owned. Even though some of the features used in the private domain classifier are not applicable (e.g., those related to apex domains), with additional content-agnostic features, we are able to achieve an accuracy of 97.2% with 97.2% precision and 98.1% recall.

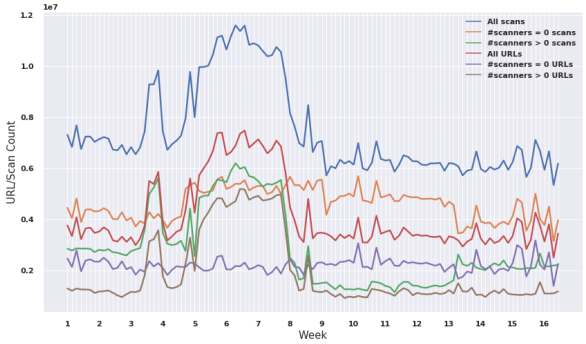


Figure 3: Daily Unique Scan and URL Counts from VT URL Feed for all, #scanners = 0) and #scanners > 0

4 VT URL Feed Dataset and Its Characteristics

In this section, we characterize and share insights into the VT URL Feed dataset, which inspires us to design some of the features used in our classifiers.

4.1 Daily Volumes

The VT URL Feed dataset contains 814,678,956 unique URLs from Aug. 1, 2019 to Nov. 18, 2019. Figure 3 shows the worldwide daily volume of unique scans and URLs in VT during our study period. Note that the same URL may be scanned multiple times in a given day. Each scan that generates a report with a new scan ID is considered a different one. However, if VT is simply queried multiple times only to retrieve an existing report instead of triggering new scans, it does not change the scan ID. Hence, such multiple reports with the same scan ID are considered as one record. It is interesting to note that the daily average of observed likely benign scans (i.e., #scanners = 0) is 89.3% of the total number of scans, which is around 4.8M. However, at the start of our study period (weeks 3-4 and weeks 5-8), we see an interesting spike in likely malicious scans and URLs (i.e., #scanners > 0). We inspect the domains marked malicious during this early period, and identify that 5 compromised domains (`gticng.com`, `clinique-veterinaire-gembloux.be`, `advancedimoveis.com`, `harikaindustries.co.in` and `cos.pt`) are used to host hundreds of thousands of malicious javascripts that resulted in the spike. Excluding these outlier domains, we observe that on average malicious URLs are scanned 6 times during the above period while benign URLs are scanned only twice. This follows the general user behavior where the more suspicious the URLs are, the more they are checked. Another interesting observation is that the daily average scan count is roughly twice the average URL count. We consider these observations when designing features for our classifiers in Section 5.

Next, we assess the coverage of malicious websites in our dataset compared to popular blacklists and reputation services.

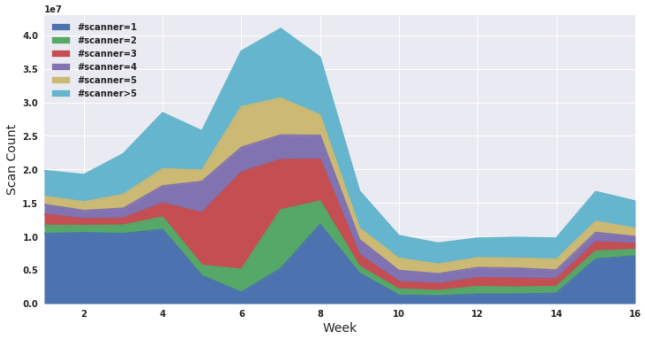


Figure 4: Weekly Unique Scan Counts with #scanners marked URLs as Malicious

Figure 4 shows the weekly distribution of #scanners counts 1, 2, 3, 4, 5, and more than 5. While there are many VT reports with 1 or 2 #scanners, on average, 45.7% among these scans have 5 or more #scanners (i.e., the top two areas in the Figure). In our work, we focus on categorizing scans with 5 or more #scanners, which corresponds to 1659K weekly malicious reports on average, or 276K malicious websites per week on average, out of which 120K are newly observed. In comparison, Google Transparency Report [17] and Phishtank [51] report around 30K and 4K per week, respectively. This shows that our study covers a much larger set of malicious websites than popular blacklists and thus has a higher impact.

4.2 Attack Types

VT scanners assign each malicious URL one of the following class labels: malicious, malware, phishing, mining, and suspicious. Since VT scanners often assign conflicting class labels, we use a simple majority voting heuristic to derive the final class label for a malicious website. We take a random sample of 100 websites of each class type and manually cross-check them against several publicly available blacklists or APIs, including Phishtank, GSB and SA. Our manual inspection showed that more than 98% of the labels using majority voting are in agreement with external intelligence, validating our heuristic. Figure 5 shows the count of attack types of malicious URLs during our study period. While malware and phishing dominate the reported malicious websites, there are only a few malicious mining and suspicious websites in our dataset. Hence, they are not shown in Figure 5. Further, malware websites are approximately 3 times more prevalent than phishing ones.

4.3 #FQDNs per Apex

Figure 6 shows the CDF of the number of FQDNs per apex during our study period for likely benign domains (i.e. #scanners = 0) and likely malicious domains (i.e. #scanners $>$ 0). Due to the highly skewed distribution, we omit the long tail of those apexes with more than 500 FQDNs. 90.2% of the

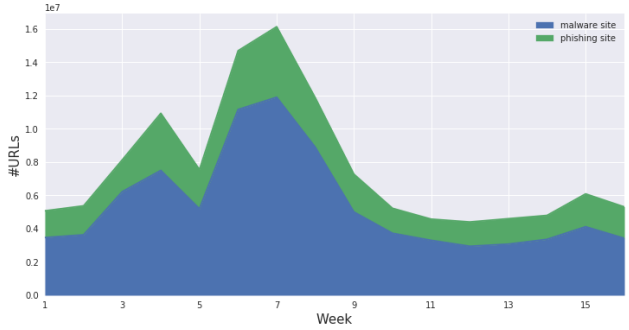


Figure 5: Attack Types

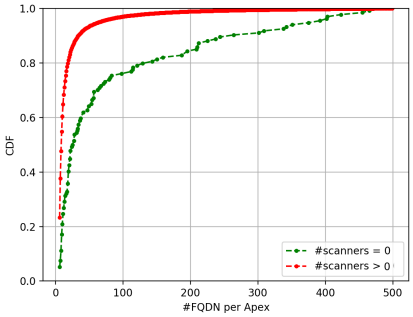


Figure 6: #FQDNs per Apex

apexes in the benign category have only one FQDN whereas only 12.3% of the apexes in the malicious category have only one FQDN. Further, as shown in Figure 6, around 40% of malicious apex domains have more than 40 FQDNs whereas only 5% of benign apex domains have more than 40 FQDNs. These observations show that attackers create many subdomains to launch their attacks in a similar fashion as fast-flux networks [36, 53]. In Section 5.2, we profile all VT reports corresponding to each apex domain and utilize the variations in the VT reports to design our compromised/attacker-owned classifier.

Another interesting observation is that there is a long tail of apex domains having more than 500 FQDNs, with some having millions. For example, `blogspot.com` (blogging), `coop.it` (URL shortener), mcafee.com (mcafee endpoint hosts) and opendns.com (Cisco open DNS) have over 1M FQDNs. We use the number of FQDNs observed as a feature in our public/private apex classifier as the higher this number is, the more likely the domain is public.

5 Construction of Classifiers

In this section, we describe the three classifiers that we design, the public/private apex classifier, the attacker-owned/compromised private apex classifier and the attacker-

owned/compromised public website classifier [1].

5.1 Public/Private Apex Domain Classifier

The goal of this classifier is to accurately predict if the apex domains of malicious URLs are public or private.

5.1.1 Ground Truth Collection

We collect a tentative public domain ground truth data set in three ways. First, we aggregate publicly available lists: the public suffix list [11], popular web hosting providers and CDN lists [5,15], and dynamic DNS lists [6,9] and take the intersection with apex domains in datasets DS1 and DS2, which results in 439 apex domains. Second, we identify potential public domains by searching over our datasets for the keywords likely to be used by public apex domains such as hosting, free, web, share, upload, drop, cdn, file, photo, and proxy. The manual inspection results in another 97 apexes. Third, we take random samples of 500 apex domains from DS1 and DS2 respectively and find additional 26 public apexes through manual inspection. Altogether, there are 562 unique public apexes across the two datasets.

We collect a tentative private domain ground truth data by randomly selecting 2000 apex domains from each dataset (DS1 and DS2) that are mutually exclusive from the tentative public dataset. We then do manual verification to create the final ground truth sets: for each apex domain, we assign a confidence score between 50 and 100 to indicate how confident we are of the label, with 100 being the most confident and 50 being undecided. To improve the quality of labeling, two domain experts performed the labeling for all the domains and we excluded the domains with conflicting labels.

During manual verification, we first check the content of the apex domain. Most of the time the content itself reveals if it is a public apex domain providing web hosting, sharing, forums or other collaborative platforms. For the remaining apexes, whose functionalities are not clear from the content, we utilize our PDNS based subdomain enumeration tool, and get the subdomains belonging to each apex domain. We then cross compare the content of these subdomains as well as their names to label the apex as public or private. For example, for public domains, different subdomains tend to have very different contents whereas for a private domain, their content follows a certain theme. With this process, we collect two ground truth sets PP-GT1 (PP stands for Public Private) and PP-GT2 from DS1 and DS2 respectively, as summarized in Table 3.

5.1.2 Feature Engineering

We extract the features in Table 4 from the VT NOD system to train a classifier. The meanings of most of the features are straightforward. Compared to private apex domains, public domains tend to host more subdomains and are scanned more frequently in VT. #subdomains and #scans capture these observations. Since subdomains are not under the control of the public apex domain owner, in practice, some of the subdomains are malicious and others are benign, whereas subdomains under private apexes tend to be mostly either benign or malicious. #Mal_Scans and Mal_Scan_Ratio capture the volume and this difference. Most public apexes, especially CDNs and proxy services, utilize FQDNs of the domains they serve (e.g. www.superwhys.com.akamai.com) whereas private apexes uses mostly descriptive popular keywords in the subdomain part such as www, mail, ns and m (for mobile). By profiling all domains seen in PDNS during the study period, we identify the top 100 subdomains as the popular keywords. We capture these differences using the #Pop_Keywords, Ratio_Pop_Keywords and #Avg_Depth features. We observe that there are more variations between subdomain names under public apex domains than those under private ones. Avg_Sub_Entropy measures the average entropy across all subdomains to capture this observation. While not directly related, #Subdomains and #Avg_Depth are inspired from public key sharing in CDNs [63], and #Pop_Keywords and #Avg_Sub_Entropy features are inspired from the diversity features described in [42].

[1] The code is available at https://github.com/qcri/compromised

Table 3: Public/Private Ground Truth

Ground truth	Public	Private
PP-GT1	410	1370
PP-GT2	528	1408

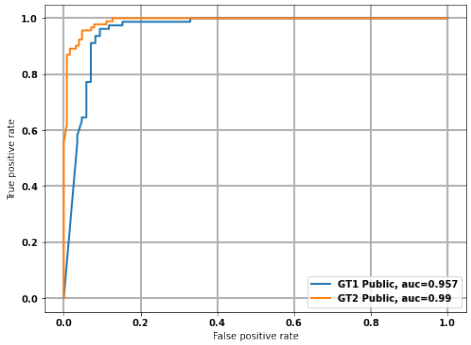


Figure 7: ROC Curves for RF Public Classifiers where Class 1 is Public

5.1.3 Model Training and Classification Accuracy

We train 8 classifiers (Support Vector Classification (SV), Random Forest (RF), Extra Tree (ET), Logistic Regression (LR), Decision Tree (DT), Gradient Boosting (GB), Ada Boosting (AB) and K-Neighbors (KN) Classification) using the features in Table 4. Out of all of the classifiers, RF performs the best.

Table 4: Public Apex Classifier Features

Feature Name	Description	Novel
VT Duration	The time between the apex domain first and last seen in VT	✓
#Scans	No. of unique scans performed for the apex and its subdomains	✓
#Mal_Scans	No. of unique scans that VT marks apex or its subdomains as malicious	✓
Mal_Scan_Ratio	The ratio of scans with malicious results and the total number of scans for apex and its subdomains	✓
#subdomains	The number of FQDNs (Fully Qualified Domain Names) observed in VT URL feed for the apex domain	[63]
#Pop_Keywords	The number of popular keywords used in the subdomain part of the FQDNs of the apex domain	[42]
Ratio_Pop_Keywords	The ratio of popular keywords used and the total number of FQDNs observed for the apex domain	✓
#Avg_Depth	The average number of subdomain levels used in the FQDNs belonging to the apex domain	[63]
Avg_Sub_Entropy	The average entropy of the subdomain parts of the FQDNs belonging to the apex domain.	[42]

Our model on both ground truth sets performs really well, showing the generalizability of our model across different datasets. With 10-fold cross validation on a balanced dataset, the RF classifier on PP-GT1 labels public apex domains with 92% accuracy, 97.4% precision and 87.5% recall. The RF classifier on PP-GT2 labels public apex domains with 97.2% accuracy, 97.7% precision and 95.6% recall. As shown in Figure 7, AUCs of the two ROC curves are 96% and 99% for GT1 and GT2 respectively, demonstrating high degrees of separability of the two classes. One reason for the better performance in GT2 is that the two classes in GT2's ground truth data have a better separation, resulting in a better decision boundaries.

5.1.4 Observations

We applied the above trained model to all the URLs in DS1 and DS2, and identified 6,675 malicious public apex domains and 725,325 malicious private apex domains in total. It is interesting to see that among all the apex domains hosting malicious URLs, only 1% are public apexes. However, these public apexes host a large portion of malicious URLs: 46.5% of malicious URLs are from public apexes. This observation is not surprising, given that attackers can utilize public apexes to deploy a large number of malicious URLs with almost no cost. Meanwhile, note that most existing work on malicious domain detection either focuses on apex domains or treats all URLs the same without distinction. Our finding suggests that malicious URLs from public apexes form a unique and significant set of Internet entities with their own distinguishing characteristics. Therefore, it would be more effective to design detection mechanisms specifically targeting such malicious URLs. Our classifier would help researchers to quickly zoom into such URLs.

Figure 8 shows the average Alexa ranking distribution for public and private apex domains. For unranked domains, we assign the insignificant rank of 1 million for better visualization. We see that public apexes have a higher average Alexa ranking than private apexes as public apex domains along with their vast number of subdomains are accessed more frequently by users. Yet it is also interesting to see that half of public domains are not popular (unranked), showing that attackers also utilize less popular public domains to launch attacks. As public apex domains could also host many benign subdomains, current registration and domain reputation based systems [29,42] and inference based systems [37,64] that rely on Alexa ranking (or domain popularity) may inadvertently blacklist public apex domains, disrupting benign sites.

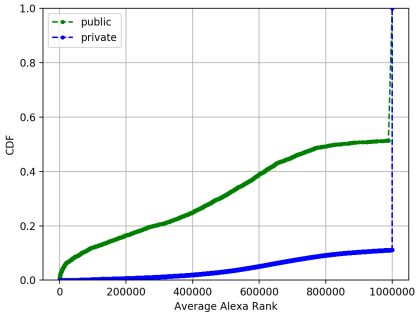


Figure 8: Average Alexa Ranking for Public and Private Apex Domains during the Study Period

5.2 Attacker-Owned/Compromised Private Apex Classifier

5.2.1 Ground Truth Collection

We manually create two ground truth sets of compromised and attacker-owned apex domains AC-GT1 (AC stands for Attacker-Owned/Compromised) and AC-GT2 from the private domains identified from DS1 and DS2 respectively using our public/private classifier.

We first select a random sample of 2500 domains from each of DS1 and DS2. We perform manual inspection of each sample and provide a confidence score to indicate how confident the domain experts are about the label. The following information and sources are manually inspected to decide if a malicious apex is compromised or attacker-owned. In addition to checking the website, we check auxiliary information such as registration information including historical WHOIS records, hosting information, and PDNS information. We also check the detailed reports from two threat intelligence platforms, `riskiq.com` and `otx.alienvault.com`. Further, we

inspect detailed reports from two reputation services, GSB and SA. To identify compromised domains, we rely on the deviations of the visual and auxiliary information in the apex domain and the domain under consideration. We observe that shadow domains, one type of compromised domains, have very different contents compared to the main website and the auxiliary information such as hosting IPs are different for the main website (reputed hosting provider) and the domain under consideration (bullet proof hosting) [42, 50]. On the other hand, attacker-owned domains have relatively new registration information [35], are likely to utilize fast flux networks [54], are short-lived (likely to be NX domain) [33], and blacklisted [39, 60]. After manual verification, we select the ones with 90% or above confidence scores assigned by the domain experts. A summary of the ground truth datasets are shown in Table 5.

Table 5: Compromised/Attacker-Owned Private Apex Ground Truth

Ground truth	Compromised	Attacker-Owned
AC-GT1	704	1004
AC-GT2	685	885

5.2.2 Feature Engineering

We identify five groups of features: lexical, VT report, VT profile, PDNS (hosting), and Alexa features. Table 6 summarizes these features. Lexical features capture the lexical properties of the URL under consideration. VT report features include those attributes that are directly available from VT reports. VT profile features are extracted from our VT NOD system. PDNS features are extracted from the Farsight Passive DNS DB system. Most of the lexical, Alexa and PDNS features either have been proposed in or adapted from previous research on detecting malicious URLs [25, 26, 34, 40, 43, 61, 66]. Past research utilizes these features to distinguish attacker-owned domains from benign domains that usually appear consistently in domain reputation lists such as Alexa Top 1M [58, 71]. In our case, these features are useful as many apexes of compromised domains are likely to have properties similar to such benign domains. We improve our classifier with additional features that collectively amplify the deviation of malicious websites hosted on benign apexes.

Next, we describe those features that either improve existing ones or are newly introduced in our work. VT Report Features are directly extracted from the VT reports. We observe that the VT_Duration feature for compromised domains tends to be higher than that for attacker-owned domains. One reason is that compromised domains are in general harder to detect by existing systems [35, 37] as attackers are exploiting the reputation of legitimate domains. Due to the same reason, we observe that the number of scanners that mark a compromised site as malicious is less than that for attacker-owned sites. Positive_count captures this observation. Compared to attacker-owned domains, we observe that attackers more often use compromised domains as a redirection site in order to evade detection, which is captured by Is_URL_Redirected.

VT profile features capture the intuition that almost all subdomains and scans of attacker-owned domains are malicious whereas only some of the subdomains and scans of compromised domains are malicious.

From the PDNS features, the number of authoritative name servers and the number of SOA domains capture the observation that attacker-owned domains change their hosting providers more often than benign domains to evade detection or takedown. Additionally, as Lever et al. [41] point out, attackers drop catch or re-register domains to exploit the residual trust in them, which also results in domain being associated with multiple name servers. Comparison of apex domains with name server domains and SOA features capture the observation that benign domains are more likely to be hosted in their own servers compared to attacker-owned ones. We improve several lexical features present in previous work [34, 43, 61]. Specifically, we observe that attacker-owned domains more often use these squatting methods to impersonate brands compared to compromised domains. We profile Alexa Top 1M domains over 1 year to identify Alexa top 1000 brands to detect combosquatting [38], levelsquatting [31] and target embedding [57] domains which are shown to be much more prevalent than traditional squatting types [27, 49]. Features Brand, Similar, and Pop_Keywords capture new squatting tactics used by attackers. The presence of these features in the apex part of domains makes a domain more likely to be an attacker-owned one. On the other hand, the presence of such lexical features in the path is likely to identify compromised ones.

5.2.3 Model Training and Classification Accuracy

We train the same 8 classifiers (SV, RF, ET, LR, DT, GB, AB and KN) as in Section 5.1.3, out of which, RF and ET performed the best.

Figure 9 shows the ROC curves for RF for both AC-GT1 and AC-GT2, with 10-fold cross validation (the ROC curves for ET are similar). Our classifier achieves 90.6% accuracy with 94.7% precision and 86.1% recall for AC-GT1, and 96.8% accuracy with 99.1% precision and 93.4% recall for AC-GT2. The fact that our model achieves high accuracy for datasets collected on different time periods shows the robustness of our approach and that it could be generalized to different ground truth datasets. Feature importance charts show that no single feature is dominant in deciding the class label which makes it difficult for adversarial manipulations.

Table 6: Attacker-Owned/Compromised Apex Classifier Features

Feature Name	Description	Novel
VT Report Features		
VT_Duration	The duration between the first and the last time the URL is scanned in VT	✓
Response_Code	The response code returned for the website as reported in VT report	[62]
Rlength	The length of the content as reported in VT report	[62]
Is_URL_Redirected	Is the final URL different from the original URL as reported in VT report?	[42]
Positive_Count	The number of scanners detected the URL as malicious	✓
Domain_Malicious	Is the domain of the URL marked as malicious in VT?	✓
VT Profile Features		
#Total_Scans	The number of times the domain is scanned earlier (extracted from VT NOD)	✓
#Benign_Scans	The number of times the domain is marked as benign earlier	✓
#Subdomain_Mal	The number of subdomains marked malicious by previous VT reports	✓
PDNS (Hosting) Features		
PDNS_Duration	The length of the domain footprint seen in PDNS	[29]
Name_Servers	The number of authoritative NS in which the domain was hosted	Derived from [41]
Query_Count	The number of lookups recorded for the domain in PDNS	[29]
SOA_Domains_Nos	The number of SOA domains under which the domain was hosted	✓
SOA_Domain	Is the apex of the domain the same as the apex of the SOA domain?	✓
Lexical Host Features		
#Subdomains	The number of levels in the subdomain part of the FQDN	[44]
Minus	The number of dashes appear in the FQDN	[44]
Brand	Does it impersonate a popular Alexa top 1000 brand?	Derived from [38]
Similar	Does the domain contain words within Levenshtein distance 2 of a popular Alexa top brand?	Derived from [38]
Fake_TLD	Does the domain name include a fake gTLD (com, edu, net, org, gov)?	Derived from [57]
Pop_Keywords	Does the domain name include popular keywords?	Derived from [38]
Entropy	The entropy of the FQDN	[24, 25]
Lexical Path Features		
Brand_In_Path	Does the path have an Alexa top 1000 brand name(s)?	Derived from [38], [45]
Similar_In_Path	Does the path contain words within Levenshtein distance 2 of a popular Alexa top brand?	Derived from [38], [45]
URL length	The length of the URL	[44, 46, 47]
#Query_Params	The number of query parameters in the URL	[40]
Alexa Features		
Alexa_Rank_Avg	The average Alexa rank during the study period	[52]
Is_In_Alexa_1Year	Does the apex appear consistently in Alexa Top 1M throughout the previous year?	Derived from [58]

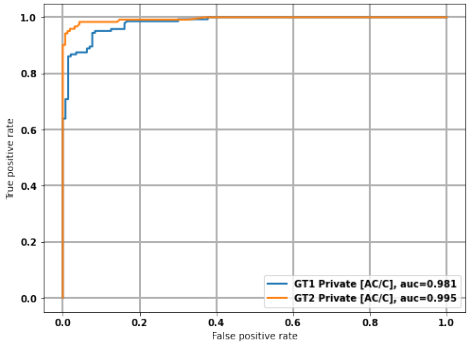


Figure 9: ROC Curves for RF Compromised/Attacker-Owned Classifiers

5.2.4 Observations

Following the pipeline shown in Figure 2, we applied the above classification model and labeled all the 725,325 private apex domains in DS1 and DS2 that host malicious URLs with #scanners ≥ 5. We observe that 65.6% of such private apex domains are classified as compromised, indicating that attackers utilize more compromised apex domains than creating their own apex domains, which could be due to several reasons. First, attackers try to ride on the reputation of compromised domains, which are also often long lived. Malicious domains deployed over compromised private apexes thus are more evasive and hard to detect by current reputation systems. Second, many private apex domains are not well maintained or patched in time. Compromised private apex domains are abundant and become more economically favorable for attackers than setting up their own apex domains, which could incur cost during domain registration.

This observation is consistent with prior work done on phishing websites [30] and public threat intelligence reports [4, 19]. Yet, our study is not limited to a specific type of malicious URLs. Instead, it covers a variety of malicious domains with a much larger scale, utilizing a more comprehensive dataset collected from VT.

Figure 10 shows the average Alexa rank distribution for compromised and attacker-owned apex domains. As expected, most of the attacker-owned domains have either low Alexa ranking or no rank. However, it is interesting to see that there are some attacker-owned domains with Alexa ranking below 100K. Another interesting observation is that about 10% of compromised private apexes are not ranked, indicating that attackers launch attacks from less popular benign websites as

well, which could be utilized to launch attacks such as DDoS that do not require reputable sites.

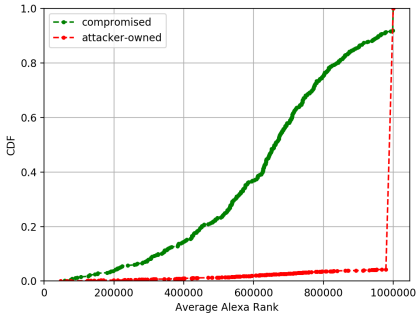


Figure 10: Average Alexa Ranking for Compromised and Attacker-Owned Domains

Figure 11 shows the number of days from the registration to first malicious behavior during our study period. The first malicious behavior is approximated as the first VT report indicating a website is malicious and thus it provides an upper bound on how soon attackers utilize these websites after registration. 20% of attacker-owned domains are utilized to launch attacks soon after they are registered. However, many other domains are utilized several months after registration, necessitating one to have detection mechanisms in place beyond the initial registration period. This behavior is consistent with the trend that attackers park their domains before using them to launch attacks so that they can evade detection by many existing reputation based systems. A concerning fact is that benign domains on the other hand gets compromised several years after they are registered. Frequent reasons for such delayed compromise are that some technologies utilized in unmaintained benign websites become outdated and/or servers on which they are deployed are not upgraded over time, making them easy targets for attackers.

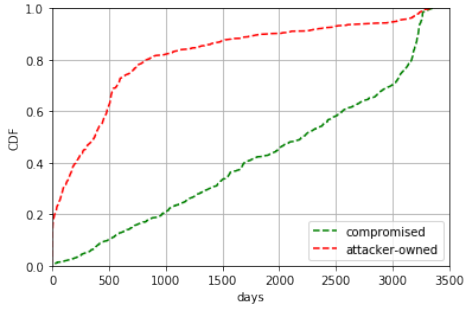


Figure 11: #Days from Registration to First Malicious Behavior During the Study Period

5.3 Attacker-Owned/Compromised Public Website Classifier

In this section, we further categorize those URLs hosted in public apexes as attacker-owned or compromised. Note that different from the classifier for private apex domains, this classifier is not to classify a public apex domain, but its subdomains that could be either prefix-based (e.g., alice.000webapphost.com) or suffix-based (e.g., sites.google.com/site/alice). For brevity, we call them public websites.

5.3.1 Ground Truth Collection

We manually create from DS1 a ground truth set of compromised and attacker-owned public websites AC-P-GT. We check the content of a public website to determine if the website is created by attackers or compromised. Further, some public apex services such as 000webapphost.com and blogspot.com identify and block some attacker-owned websites. We use this information to collect additional attacker-owned public websites. In total we collect 613 compromised public websites and 1157 attacker-owned public websites.

5.3.2 Feature Engineering

We utilize all features in Table 6 except the hosting features Name_Servers, SOA_Domains_Nos and SOA_Domain as public websites from a given apex domain often have similar hosting infrastructures managed by the apex domain owner. It should be noted that unlike in the private apex classifier, features are extracted at subdomain or path suffix level as our goal is to classify public websites, not apexes. Further, we utilize the additional path features listed in Table 7. We noticed that most long lived public websites (like blogs) have many associated pages (URLs), but attacker-created ones are usually short lived and tend not to create many pages to launch their attacks. The feature #URLs captures this difference. Variations in the paths belonging to a given public website are captured by features Std_Path_Depth and Std_Query_Params, as compromised public websites are likely to create paths quite different from those created by attackers.

5.3.3 Model Training and Observations

We train a RF classifier with a balanced dataset. We achieve an accuracy of 97.2% with 97.2% precision and 98.1% recall with 10-fold cross validation. Figure 12 shows the ROC curve for this classifier.

We then utilize this classifier to label the remaining public websites in DS1. We observe that, unlike private apexes, attackers primarily create their own subdomains or path prefixes on public domains (80.5%). We attribute this difference to the low cost associated with creating public websites.

Table 7: Additional features for the public website classifier

Feature Name	Description	Novel
#URLs	The number of URLs corresponding to the website.	Derived from [52]
Std_Path_Depth	Standard deviation of the path depth of URLs belonging to the website.	✓
Std_Query_Params	Standard deviation of the number of query parameters for each URL belonging to the website.	✓

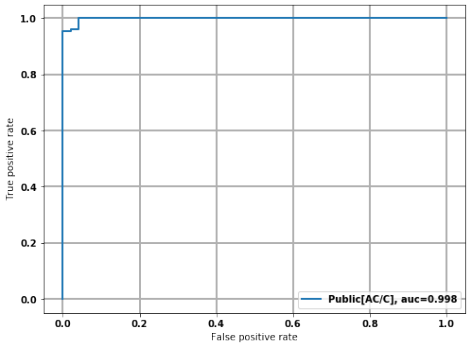


Figure 12: ROC Curve for RF Public Compromised/Attacker-Owned Classifier

We analyze the Alexa ranking associated with the identified attacker-owned/compromised public websites. As expected, only a small fraction (2.28%) of public malicious websites in DS1 made it to Alexa top 1M during the attack time period. However, it is interesting to observe that during this time more compromised public websites (6.87%) are observed in Alexa top list compared to 1.17% of attacker-owned websites. Further, compromised ones stayed in the Alexa top list longer than attacker-owned ones (6.8 vs. 2.9 days). These observations indicate that even though compromised public websites are the minority, the damage they may cause is higher than that of attacker-owned public websites.

5.4 Summary of Attack Types

Table 8 summarizes the distribution of attack types classified by our two-step classification of URLs. In the first step, we classify apex domains as public and private. In the second step, we classify private apexes as compromised and attacker-owned apexes, and websites of public apexes as compromised and attacker-owned, which include both prefix based subdomains and suffix based paths.

Table 8: Distribution of Attack Types in our Dataset

Type	Public	Private
Malicious	1% Apexes 46.5% URLs	99% Apexes 53.5% URLs
Compromised	20.5% Sites	65.6% Apexes
Attacker-Owned	79.5% Sites	34.4% Apexes

6 Classifier Analysis

We have shown so far the features of the three classifiers and their classification accuracy over the malicious URL datasets collected from VT. In this section, we perform further analysis of their properties, including how well they could be generalized to URL intelligence beyond VT, their robustness against feature manipulation, the impact when the training data are noisy or of a smaller scale, and how they compare with current industrial practices. Due to space limit, we focus our analysis on the classifier that classifies private apex domains as compromised or attacker-owned (see Section 5.2).

6.1 Applicability to Other URL Intelligence Sources

Our discussion so far is based on the data collected from VT. Indeed some of the features for the private apex classifier are also specific to VT. It raises the question whether our approach could be applied with other URL intelligence sources. In this section, we show how our methodology can be adapted to work with other intelligence feeds. In particular, we adapt our approach to build a private apex classifier over the data from Phishtank. Phishtank makes a verified list of phishing URLs every hour. We take the list of URLs appeared in our second study period from Nov 1 2019 to Nov 14 2019.

Ground Truth Collection: We collect 7756 URLs from Phishtank during the study period. First we filter the public apex domains by passing the URLs through our public/private classifier. This results in 6377 private phishing URLs and 2804 private apex domains. Following a process similar to the GT collection for the private AC/C classifier, out of the 2804 private apex domains, we collect 183 compromised domains and 392 attacker-owned domains.

Feature Extraction: We collect all features except VT profile features mentioned in Table 6, as they are specific to VT and are not appliable for Phishtank.

Model Training: Similar to other classifiers, we train a RF classifier with a balanced dataset. We achieve an accuracy of 91.2% with 93.5% precision and 93.5% recall, which is comparable to the accuracy achieved over the VT dataset. Figure 13 shows the ROC curve for this classifier.

The performance is slightly lower than that for the private AC/C classifier for VT URLs. We attribute this difference to smaller dataset sizes as well as the reduced number of features utilized. We believe the performance could be improved by utilizing additional features such as registration information and certificate information.

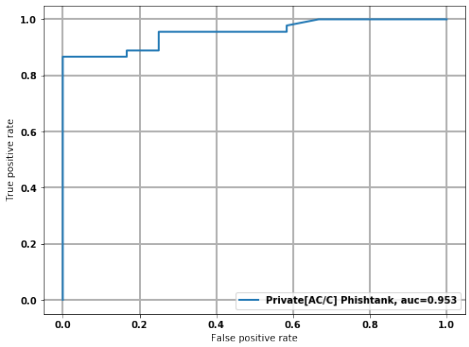


Figure 13: ROC Curve for RF Private Compromised/Attacker-Owned Classifier for Phishtank URLs

6.2 Robustness

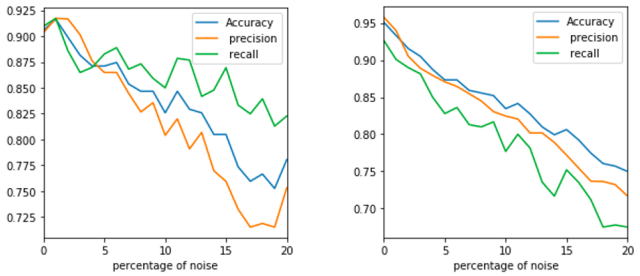
As VT provides services to the public, it could be a concern that attackers may submit URL queries and indirectly influence the VT features, e.g., #Total_Scans and #VT_Duration, and consequently the classification results in their favor. To show the classifier's robustness against such manipulations, we measure the performance of the classifier when different types of VT features are excluded. As shown in Table 9, the influence of these features on the classification performance is not significant. Even when we aggressively omit all VT features, the classification accuracy drops by only 6%. A possible way to further improve robustness is to enrich the classifier with additional features from disparate sources such as domain certificates.

Table 9: Robustness of Private AC/C Classifier

Features	Acc.	Prec.	Rec.
All	96.4%	99.1%	92.6%
All - {VT Profile}	94.01%	94.1%	91.8%
All - {VT Profile,VT Duration, Positive Count}	92.9%	93.9%	90.9%
All - {VT Profile, VT Report}	90.1%	92.0%	84.4%

6.3 Impact of Training Data Quality

The effectiveness of machine learning depends greatly on the quality of the training data. In our study, we collect labeled training and testing data through manual inspection by multiple domain experts and adopt mechanisms to handle disagreements. Here we would like to see how our classifier would be affected if the training dataset is noisy, i.e., with some data mislabeled. For this purpose, we deliberately inject mislabeled training data, and re-train our classifier for both DS1 and DS2, while controlling the noise level, i.e., the percentage of mislabeled training data. As shown in Figure 14, in general our classifier can tolerate small amount of mislabeled data. At 1% and 5% noise levels, the accuracy of our classifier is reduced by only 0.9% and 4.2% respectively. Further, there seems to be a linear correlation with the noise level and the classifier accuracy. When a significant portion of the training data is mislabeled (e.g., 15%), the classifier accuracy drops greatly.



(a) AC-GT1 (b) AC-GT2

Figure 14: Performance with Noisy Labels

6.4 Impact of the Size of Training Data

An important question in machine learning model is to identify how much training data is sufficient to achieve the desired performance. Figure 15 shows the accuracy of our private apex domain classifier for different dataset sizes. Recall that the size of our original balanced dataset for the two windows is approximately 700 apexes from each class. As shown in this figure, our classifiers yield an accuracy similar to the full labeled set with approximately 70% of the labeled data.

6.5 Feature Stability over Time

Identifying how often one needs to re-train a classifier to cope with concept drift is quite important in practice. To measure the impact of concept drifting on our classifier, we create two datasets which are one and two weeks apart from the AC-GT2 dataset. We evaluate the performance of our classifier trained on AC-GT2 with these two datasets of 100 labels that are one and two weeks away from the training set. As shown in Table 10, our classifier maintains a good performance after two weeks, though it is also clear its performance drops gradually as the temporal gap between the training and testing data increases. In order to maintain a high precision, we recommend to retrain the classifiers weekly.

We further use the model trained with the labeled data in AC-GT1 to classify data in AC-GT2. As expected, since the two datasets are two months apart temporally, the classification accuracy drops dramatically, by 14%.

Table 10: Concept Drift Analysis of Private AC/C Classifier

Validation Set	Acc.	Prec.	Rec.
Same Week	97.1%	99.1%	94.2%
After 1 Week	95.0%	90.9%	100.0%
After 2 Weeks	93.0%	87.7%	100.0%

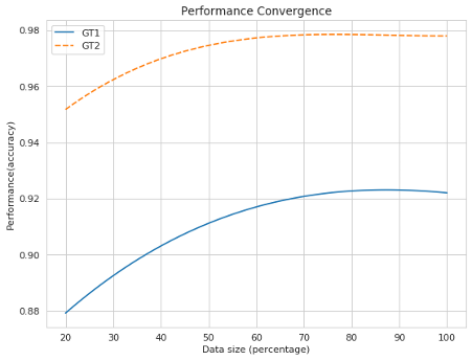


Figure 15: Accuracy of the Model with respect to the size of the dataset

6.6 Misclassified Apex Domains

We utilize LIME [56], a well-known tool that provides explanations for individual predicted data points, in order to study the misclassified data points in AC-GT2. This classification results in 1 False Positive (FP) and 8 False Negatives (FNs). We make two observations from this analysis. First, most of the misclassified data points do not have PDNS features (we use default values for missing PDNS features). Second, the probability of prediction for the rest of the misclassified ones is close to 0.5 making the prediction weak. Possible approaches to further reduce FPs/FNs are to either fill missing values using another similar data source such as active DNS and/or incorporate additional features from desperate sources such as WHOIS registration records in order to differentiate the two classes further.

6.7 Comparison with Industry Practices

GSB [16] has been instrumental in protecting users across the web from phishing and malware attacks. GSB is integrated with several browsers including Chrome and also provides API based access. GSB categorizes malicious websites as either malware sites or phishing sites. Malware sites are further classified as compromised or attacker-owned sites. However, GSB does not provide public APIs or services that directly classify individual URLs as compromised or attacker-owned. Instead, it only reports aggregated statistics of these two types of URLs that GSB has discovered. There is also no document or paper detailing exactly how GSB classifies compromised and attacker-owned domains. Therefore, we could not directly compare our classifier with that used by GSB. Here we compare the published statistics of these two types of URLs in the Google Transparency Report [17] in August 2019. Figure 16 compares the statistics on the number of unique malicious websites detected by GSB and VT.

While GSB detects around 30K new malicious websites per week, VT detects 3 times more than that amount, which shows that there is room to improve the coverage of malicious domains of GSB. Our manual inspection of selected malicious

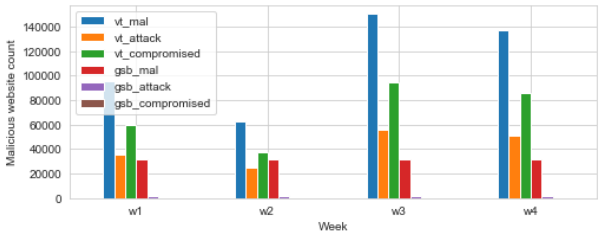


Figure 16: Comparison of GSB and Our Approach

websites from VT confirmed this observation, i.e., there exists many malicious domains marked by VT but not by GSB. Further, from the Google Transparency Report, we see that GSB only studies whether malware websites are compromised or attacker-owned, yet, malware websites (2K websites on average per week during the comparison time period) only account for less than 7% of all the malicious websites detected by GSB. In comparison, we categorize both phishing and malware websites as attacker-owned or compromised. We believe our approach can complement GSB to automatically detect more attacker-owned/compromised domains.

In the APWG 2016 phishing trends report, Aaron et al. [21] proposes to utilize three heuristics to distinguish compromised domains from attacker-owned ones. They flag a domain as malicious if it is reported for phishing within a very short time of being registered, and/or contains a brand name or misleading strings, and/or is registered in a batch or in a pattern that indicates common ownership or intent. While such a heuristic based approach may accurately identify some attacker-owned/compromised domains, our analysis shows that it misclassifies many malicious domains. During our study period we observe that 13% of attacker-owned domains are detected after 3 months and they do not have any brand names. In their approach, these domains are likely to be misclassified as compromised.

7 Limitations and Future Work

Features specific to URL intelligence sources. Our work primarily utilizes the malicious URL intelligence from VT. Indeed some of the features are specific to VT reports. We have shown that even when such features are removed, our classifiers could still perform well. Further, through experiments over the Phishtank dataset, we also show that our classifiers could be adapted to work with other URL intelligence sources. However, admittedly, the accuracy on the Phishtank dataset is not as high as that on the VT dataset with VT specific features. The observation is that, though our approach is general enough, data source specific features would bring additional improvement to our classifier. Thus, in practice, when applying our classifier with other URL data sources, it pays to derive further URL data source dependent features to enhance our model. Similarly, we derived features utilizing other data sources such as PDNS and Alexa domain ranking.

We did not explore other publicly available data sources, e.g., WHOIS registration records, active DNS records, and certificate transparency logs. It is possible to design additional features from such data sources to further improve our model. Another promising direction is to utilize content based classification as the second layer of categorization of websites whose predicted label is close to the decision boundary, i.e. the probability of prediction is close to 50%. Such an approach scales to millions of URLs as content analysis, which is resource-intensive, is performed only on a fraction of them.

Ground Truth. It is always a challenge to collect high-quality ground truth training data for machine learning tasks. It is particularly so for malicious domain research. In this paper, we obtain through manual inspection labeled datasets for training and testing, which is inevitably a tedious and time-consuming process. As a result, our labeled data set is only of a moderate scale (ranging from a few hundreds to over a thousand). It is certainly desirable to evaluate our models on a much larger data set, which could shed new insights of our approach. In this work, we did not explore ways to obtain labeled datasets through automated or semi-automated processes. However, as shown in Section 6.3, noisy labeled data tend to impact the accuracy of the trained model, especially when mislabeled data account for a non-negligible portion of the training data. How to balance the scale and quality of training data, through advanced machine learning techniques (e.g., weakly supervised technique such as Snorkle [55]) is an interesting and important problem for malicious domain research.

Re-Compromised Websites. We inspect random samples of compromised domains predicted by our classifier and retrospectively analyze them utilizing historical VT reports. We find a concerning trend that some compromised websites after being cleaned, which is indicated by subsequent VT clean reports, gets compromised again. One possible reason for such behavior is that an underlying vulnerability still remains. A useful future direction is to come up with a reputation based score for benign websites based on how often they get compromised and how quickly identified infections are cleaned.

8 Related Work

Malicious vs. compromised domains. Moore et al. [48] show how Internet miscreants utilize Google search to identify vulnerable web servers that use unpatched software and host phishing web pages. They also show how such servers get repeatedly compromised when the root cause of vulnerability is not addressed. They assess that 75.8% of the phishing web sites they analyzed are hosted on compromised web servers. Corona et al. [30] proposes an approach to detecting phishing websites hosted on compromised domains by comparing the HTML code and visual appearance of potential phishing pages against the corresponding characteristics of the homepage of compromised (hosting) website. Recently, Sophie et al. [52] build a content-agnostic machine learning model using three different phishing datasets APWG [3] and PhishLabs [10] and DeltaPhish [30]. However, there are several shortcomings in their work: their classifier heavily relies on The Wayback Machine (WBM) [13] features that are not only biased but also difficult to collect. We observe that WBM content is not available for many attacker-owned domains, non-US websites as well as newly registered domains, leading many missing values in feature vectors. Further, some of their predicted labels with high confidence are in fact inaccurate (e.g. 000a.biz and kl.com.ua are public hosting domains). An interesting approach to identifying compromised domains has been proposed by Liu et al. [42]. There key ideas to profile the good behavior of the passive DNS information of each domain and measure the deviation as a differentiator. However, their approach fails to accurately filter public domains and additionally the classification requires considerable reputation information on domains in order to make an accurate decision. Recently, Maroofi et al. [45] proposed a content based approach to classify malicious domains as attacker-owned or compromised. They extract features from WHOIS registration records, passive DNS, active DNS, page ranking formation, and page content. Similar to previous approaches, their approach focuses only on private apex domains. Further, they filter public domains based only on the publicly available suffix lists. Yet we show in this work that such lists cover only a small fraction of public domains. This results in inaccurate classification as most characteristics of public apex domains are different from private apex domains. Further, unlike our approach, all above approaches ignore compromised websites on public domains.

Domain impersonation attacks. Malicious domains are increasingly known to use cybersquatting such as combosquatting [38] and target embedding [57] techniques to trick more victims by mimicking legitimate domains and embedding known popular "brand" names such as `paypal` or `apple` in the domain name. While many of such domains are attacker created, there are notable exceptions as long as they use brand keywords in good faith (e.g. applefarm.com and amazonkeratin.com). Hence, relying solely on likely brand impersonation could result in many false positives. In our work, we utilize brand impersonation as a likely signal of attacker-owned domains, but it works together with other features to improve the detection accuracy.

Phishing/malicious domain detection. These methods can broadly be categorized into two groups: content-based and content-agnostic. Content-based phishing for example [65,70] utilizes features from the web page content itself to train a machine learning model to detect phishing URLs. While they are quite accurate, it is quite time consuming and resource intensive to train classifiers based on the content of web pages. Content-agnostic phishing methods, on the other hand, utilize features other than content based features such as URL/domain lexical features, registration information, DNS

information and hosting information [26, 61]. All these methods are in fact utilize features indicative of attacker-owned domains (e.g. newly registered, hosted on an unreputed infrastructure, and fast IP fluxing) and hence perform poorly detecting compromised domains.

9 Conclusions

We design machine learning models to distinguish two kinds of malicious URL hosting apex domains, public and private. This classification helps security professionals specify which domain levels to block, the whole apex domain in the case of private apexes or specific subdomains/path suffixes in the case of public ones. Our results show that we can classify apex domains as public or private with 97.2% accuracy, 97.7% precision and 95.6% recall. From the private malicious domains, we also design another machine learning model to differentiate attacker-owned from compromised hosting apexes. This distinction is crucial to help security operators take the appropriate mitigation actions. For example, attacker-owned domains could be blocked permanently whereas compromised ones temporarily. The result shows that this classifier achieves 96.8% accuracy with 99.1% precision and 93.4% recall. We also design a classifier with high accuracy to classify public websites as attacker-owned or compromised. In terms of statistics, our results reveal a concerning trend of the malicious domains observed from VT URL Feed: most of the attacks are launched from websites whose apexes are not owned by attackers. Even though public apex domains are less than 1% of the apexes hosting malicious websites, they amount to a whopping 46.5% malicious web pages seen in VT URL feed during our study period. Out of the remaining websites (53.5%), we observe that attackers mostly compromise benign websites (65.6%) to launch their attacks, whereas only 34.4% of malicious websites are hosted on domains created by attackers. Understandably, public malicious websites exhibit the opposite trend where most (79.5%) are attacker owned. The key insight here is that more has to be done by legitimate domain owners to prevent miscreants from misusing their domains to launch stealthy attacks.

References

[1] Google Safe Browsing. https://developers.google.com/safe-browsing. Accessed: 10-08-2020.

[2] Anti-Cybersquatting Consumer Protection Act. https://icannwiki.org/Anti-Cybersquatting_Consumer_Protection_Act, 2019. Accessed December 2019.

[3] Anti-Phishing Working Group. https://apwg.org, 2019.

[4] APWG Phishing Trends Reprot Q2. https://docs.apwg.org/reports/apwg_trends_report_q2_2019.pdf, 2019. Accessed December 2019.

[5] CDN Planet CDN List. https://www.cdnplanet.com/cdns/, 2019. [Online; accessed 25-October-2019].

[6] DNS Lookup Dynamic DNS List. https://dnslookup.me/dynamic-dns/, 2019. [Online; accessed 25-October-2019].

[7] ICANN Domain Abuse Procedure. https://go.icann.org/3lS1eM1, 2019. Accessed December 2019.

[8] Netcraft Site Take Down Service. https://netcraft.com, 2019. Accessed December 2019.

[9] Neu5ron Dynamic DNS List. https://gist.github.com/neu5ron/860c158180e01b61a524, 2019. [Online; accessed 25-October-2019].

[10] PhishLabs. https://phishlabs.com, 2019. Accessed December 2019.

[11] Public Suffix List. https://publicsuffix.org/, 2019. [Online; accessed 10-February-2019].

[12] Site Take Down Service. https://sitetakedown.com, 2019. Accessed December 2019.

[13] The Internet Wayback Machine. https://www.archive.org, 2019. Accessed December 2019.

[14] Verizon Data Breach Report. https://enterprise.verizon.com/resources/reports/dbir/, 2019. Accessed December 2019.

[15] WPO Foundation CDN List. https://github.com/WPO-Foundation/webpagetest/blob/master/agent/wpthook/cdn.h, 2019. [Online; accessed 25-October-2019].

[16] Google Safe Browsing: Making the world's information safely accessible. https://safebrowsing.google.com, 2020. Accessed September 2020.

[17] Google Transparency Report. https://transparencyreport.google.com, 2020. Accessed September 2020.

[18] McAfee Site Advisor. https://www.mcafee.com/siteadvisor, 2020. Accessed January 2020.

[19] Microsoft Security Intelligence Report. https://info.microsoft.com, 2020. Accessed January 2020.

[20] Microsoft SmartScreen. https://www.microsoft.com/en-us/edge, 2020. Accessed January 2020.

[21] Greg Aaron and Rod Rasmussen. APWG Global Phishing Survey: Trends and Domain Name Use in 2016. https://docs.apwg.org/reports/APWG_Global_Phishing_Report_2015-2016.pdf, 2016. Accessed October 2020.

[22] Alexa. Alexa Top Sites. https://alexa.com. Accessed: 10-01-2021.

[23] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, II, and D. Dagon. Detecting Malware Domains at the Upper DNS Hierarchy. In *USENIX*, pages 27–27, 2011.

[24] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From throw-away traffic to bots: Detecting the rise of dga-based malware. In *Presented as part of the 21st USENIX Security*, pages 491–506, Bellevue, WA, 2012. USENIX.

[25] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González. Classifying phishing urls using recurrent neural networks. In *eCrime*, pages 1–8, 2017.

[26] A. C. Bahnsen, U. Torroledo, D. Camacho, and S. Villegas. Deepphish: Simulating malicious AI. In *2018 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–8, 2018.

[27] A. Banerjee, Md S. Rahman, and M. Faloutsos. SUT: Quantifying and Mitigating URL Typosquatting. *Computer Networks*, 55(13):3001 – 3014, 2011.

[28] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel. Exposure: A passive dns analysis service to detect and report malicious domains. *ACM TISS*, 16(4):14:1–14:28, April 2014.

[29] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel. Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains. *ACM TISS*, 16(4):14:1–14:28, apr 2014.

[30] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu, G. Mureddu, D. Ariu, and F. Roli. Deltaphish: Detecting phishing webpages in compromised websites. In *ESORICS*, pages 370–388, 2017.

[31] K. Du, H. Yang, Z. Li, H. Duan, S. Hao, B. Liu, Y. Ye, M. Liu, X. Su, G. Liu, Z. Geng, Z. Zhang, and Jinjin Liang. Tl;dr hazard: A comprehensive study of level-squatting scams. In *SPCN*, pages 3–25, 2019.

[32] Farsight Security, Inc. DNS Database. https://www.dnsdb.info/. Accessed: 10-01-2021.

[33] Pawel Foremski. The modality of mortality in domain names an indepth study of domain lifetimes. In *Virus Bulletin Conference*, 2018.

[34] S. Garera, N. Provos, M. Chew, and A. D. Rubin. A framework for detection and measurement of phishing attacks. In *CCS*, pages 1–8, 2007.

[35] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster. PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration. In *CCS*, pages 1568–1579, 2016.

[36] C.-H. Hsu, C.-Y. Huang, and K.-T. Chen. Fast-flux Bot Detection in Real Time. In *RAID*, pages 464–483, 2010.

[37] I. M. Khalil, B. Guan, M. Nabeel, and T. Yu. A domain is only as good as its buddies: Detecting stealthy malicious domains via graph inference. In *CODASPY*, pages 330–341, 2018.

[38] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *CCS*, pages 569–586, New York, NY, USA, 2017. ACM.

[39] Marc Kührer and Thorsten Holz. An Empirical Analysis of Malware Blacklists. *Praxis der Informationsverarbeitung und Kommunikation*, 35(1):11–16, 2012.

[40] Anh Le, Athina Markopoulou, and Michalis Faloutsos. Phishdef: Url names say it all. *2011 Proceedings IEEE INFOCOM*, pages 191–195, 2011.

[41] C. Lever, R. Walls, Y. Nadji, D. Dagon, P. McDaniel, and M. Antonakakis. Domain-Z: 28 Registrations Later Measuring the Exploitation of Residual Trust in Domains. In *Proceedings of the IEEE SP*, pages 691–706, 2016.

[42] D. Liu, Z. Li, K. Du, H. Wang, B. Liu, and H. Duan. Don't let one rotten apple spoil the whole barrel: Towards automated detection of shadowed domains. In *Proceedings of the 2017 ACM CCS*, CCS '17, pages 537–552, New York, NY, USA, 2017. ACM.

[43] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious urls. In *Proceedingsof theSIGKDD Conference. Paris,France*, 2009.

[44] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious urls. In *Proceedingsof theSIGKDD Conference. Paris,France*, 2009.

[45] Sourena Maroofi, Maciej Korczynski, Cristian Hesselmanz, Benoıt Ampeaux, and Andrzej Duda. COMAR: Classification of compromised versus maliciously registered domains. In *IEEE EuroS&P*, pages 1–14. IEEE, 2020.

[46] D. K. McGrath and M. Gupta. Behind phishing: An examination of phisher modi operandi. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[47] D. Kevin McGrath and Minaxi Gupta. Behind phishing: An examination of phisher modi operandi. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, LEET'08, USA, 2008. USENIX Association.

[48] T. Moore and R. Clayton. Fc. chapter Evil Searching: Compromise and Recompromise of Internet Hosts for Phishing, pages 256–272. 2009.

[49] N. Nikiforakis, S. Van A., W. Meert, L. Desmet, F. Piessens, and W. Joosen. Bitsquatting: Exploiting Bit-flips for Fun, or Profit? In *WWW*, pages 989–998, 2013.

[50] A. Noroozian, J. Koenders, E. Van Veldhizen , C. H. Ganan, S. Alrwais, D. McCoy, and M. Van Eeten. Platforms in everything: Analyzing ground-truth data on the anatomy and economics of bullet-proof hosting. In *USENIX*, pages 1341–1356, 1 2019.

[51] OpenDNS. PhishTank. https://www.phishtank.com/. Accessed: 16-02-2019.

[52] S. L. Page, G. Jourdan, G. v. Bochmann, I. Onut, and J. Flood. Domain classifier: Compromised machines versus malicious registrations. In *ICWE*, pages 265–279, 2019.

[53] E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi. FluXOR: Detecting and Monitoring Fast-Flux Service Networks. In *DIMVA*, pages 186–206, 2008.

[54] R. Perdisci, I. Corona, D. Dagon, and Wenke Lee. Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces. In *ACSAC*, pages 311–320, 2009.

[55] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow.*, 11(3):269–282, November 2017.

[56] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD*, KDD '16, page 1135–1144, 2016.

[57] R. Roberts, Y. Goldschlag, R. Walter, T. Chung, A. Mislove, and D. Levin. You are who you appear to be: A longitudinal study of domain impersonation in tls certificates. In *CCS*, pages 2489–2504, 2019.

[58] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez. A long way to the top: Significance, structure, and stability of internet top lists. In *IMC*, page 478–493, 2018.

[59] M. Sharif, J. Urakawa, N. Christin, A. Kubota, and A. Yamada. Predicting impending exposure to malicious content from user behavior. In *CCS*, page 1487–1501, 2018.

[60] Steve Sheng, Brad Wardman, Gary Warner, Lorrie Faith Cranor, Jason Hong, and Chengshan Zhang. An Empirical Analysis of Phishing Blacklists. In *Proceedings of the Sixth Conference on Email and Anti-Spam*, 2009.

[61] H. Shirazi, B. Bezawada, and I. Ray. "kn0w thy doma1n name": Unbiased phishing detection using domain name based features. In *SACMAT*, pages 69–75, 2018.

[62] G. Stringhini, C. Kruegel, and G. Vigna. Shady paths: Leveraging surfing crowds to detect malicious web pages. In *CCS*, page 133–144, 2013.

[63] F. Stringhlosi, T. Chung, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. Measurement and analysis of private key sharing in the https ecosystem. In *CCS*, page 628–640, 2016.

[64] X. Sun, M. Tong, J. Yang, L. Xinran, and L. Heng. Hindom: A robust malicious domain detection system based on heterogeneous information network with transductive classification. In *22nd RAID*, pages 399–412, 2019.

[65] Ke Tian, Steve T. K. Jan, Hang Hu, Danfeng Yao, and Gang Wang. Needle in a haystack: Tracking down elite phishing domains in the wild. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, pages 429–442, New York, NY, USA, 2018. ACM.

[66] R. Verma and K. Dyer. On the character of phishing urls: Accurate and robust statistical learning classifiers. In *Proceedings of the 5th ACM CODASPY*, pages 111–122, New York, NY, USA, 2015. ACM.

[67] VirusTotal, Subsidiary of Google. VirusTotal – Free Online Virus, Malware and URL Scanner. https://www.virustotal.com/. Accessed: 04-05-2016.

[68] L. Wang, A. Nappa, J. Caballero, T. Ristenpart, and A. Akella. Whowas: A platform for measuring web deployments on iaas clouds. In *Proceedings of the 2014 IMC*, page 101–114, 2014.

[69] Florian Weimer. Passive DNS Replication. In *FIRST*, page 98, 2005.

[70] Colin Whittaker, Brian Ryner, and Marria Nazif. Large-Scale Automatic Classification of Phishing Pages. In *NDSS '10*, 2010.

[71] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier. A survey on malicious domains detection through dns data analysis. *ACM Comput. Surv.*, 51(4), July 2018.