Discovering HTTPSified Phishing Websites Using the TLS Certificates Footprints

Yuji Sakurai*, Takuya Watanabe†, Tetsuya Okuda†, Mitsuaki Akiyama†, and Tatsuya Mori*‡

*Waseda University
†NTT Secure Platform Laboratories
‡National Institute of Information and Communications Technology
Email: {s5i, watanabe, mori}@nsl.cs.waseda.ac.jp, tetsuya.okuda.uy@hco.ntt.co.jp, akiyama@ieee.org

***Abstract*—With the recent rise of HTTPS adoption on the Web, attackers have begun "HTTPSifying" phishing websites. HTTPSifying a phishing website has the advantage of making the website appear legitimate and evading conventional detection methods that leverage URLs or web contents in the network. Further, adopting HTTPS could also contribute to generating intrinsic *footprints* and provide defenders with a great opportunity to monitor and detect websites, including phishing sites, as they would need to obtain a public-key certificate issued for the preparation of the websites. The potential benefits of certificate-based detection include (1) the comprehensive monitoring of all HTTPSified websites by using certificates immediately after their issuance, even if the attacker utilizes dynamic DNS (DDNS) or hosting services; this could be overlooked with the conventional domain-registration-based approaches; and (2) to detect phishing websites before they are published on the Internet. Accordingly, we address the following research question: *How can we make use of the footprints of TLS certificates to defend against phishing attacks?* For this, we collected a large set of TLS certificates corresponding to phishing websites from Certificate Transparency (CT) logs and extensively analyzed these TLS certificates. We demonstrated that a *template* of common names, which are equivalent to the fully qualified domain names, obtained through the clustering analysis of the certificates can be used for the following promising applications: (1) The discovery of previously *unknown* phishing websites with low false positives and (2) understanding the infrastructure used to generate the phishing websites. We use our findings on the abuse of free certificate authorities (CAs) for operating HTTPSified phishing websites to discuss possible solutions against such abuse and provide a recommendation to the CAs.**

1. INTRODUCTION

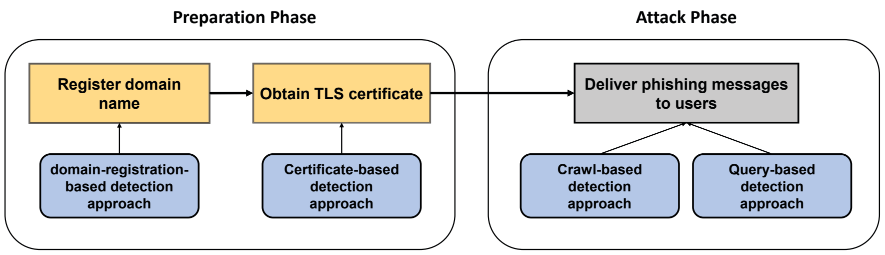
The adoption of HTTPS on the Web has increased drastically over the past few years [1], [2]. According to Google's Transparency Report [1], in several countries, such as the United States, Germany, and France, more than 90% of Web traffic has been "HTTPSified." The rate of HTTPSified Web traffic in other countries has also grown over time; for example, in Brazil, Japan, and India, more than 70% of the Web traffic has been encrypted with HTTPS. The primary factors that contribute to the drastic increase in the adoption of HTTPS are continuing HTTPS promotion efforts, such as changes in search engine rankings [3], revisions to security indicators on Web browsers [4], [5], and the publication of useful tools to install or assess HTTPSified websites [6], though the outreach of HTTPS could be widened to impact several other areas [2]. Notably, the cost of the "S" in HTTPS has been significantly reduced in recent times, as reported by Naylor et al. [7]. These changes should have contributed to the widespread adoption of HTTPS.

However, even as the number of HTTPSified websites has drastically increased, phishing websites have also started adopting HTTPS. By adopting HTTPS, an attacker could make his/her phishing website appear legitimate. In addition, the end-to-end encryption mechanism ensures that access to the HTTPSified phishing website can evade network-level detection (e.g., at a web proxy or gateway) that leverages URLs or web content. Furthermore, the recent rise in freely available certificate authorities (CAs), such as Let's Encrypt [6] and cPanel [8], has lowered the barriers to deploying HTTPS on a website. According to the 2019 Q1 Phishing Activity Trends Report of the Anti-Phishing Working Group (APWG) [9], less than 2% of phishing websites in 2015 adopted HTTPS; however, this number started increasing rapidly since the end of 2016, and reached 74% in 2019.

While HTTPSifying a phishing website may bring several advantages for an attacker, it could also contribute to generating intrinsic *footprints*, which in turn be used to systematically detect the HTTPSified phishing website. The key insight behind this assumption is that by HTTPSifying a website, an attacker must register a valid public-key certificate (i.e., TLS certificate) that contains intrinsic features such as issued date, issuer name (CA), and common name (CN). The CN in a certificate is equivalent to the fully qualified domain name (FQDN) of a server. In addition, certificate transparency (CT), which is a standardized framework to publish the public logs of all the issued TLS certificates, plays a vital role for monitoring and auditing TLS certificates. Thus, we expect that we can efficiently detect phishing websites by analyzing TLS certificates.

To understand phishing-detection approaches in terms of a phase in a phishing-attack process, we classified the approaches into domain-name registration, obtaining of

DOI 10.1109/EuroS&PW51379.2020.00076



Preparation Phase
Attack Phase
Register domain name
Obtain TLS certificate
Deliver phishing messages to users
domain-registration-based detection approach
Certificate-based detection approach
Crawl-based detection approach
Query-based detection approach

Figure 1. Comparison of phishing-detection approaches based on domain registration, certificate, crawl, and query by their detection phase.

issued certificates, and phishing-message delivery to users as shown in Fig. 1. The first two approaches comprise the preparation phase, and the last approach comprises the attack phase. The detection approaches in the attack phase, i.e., crawl- and query-based approaches, can find phishing websites only after they are published because these approaches are triggered by the delivered phishing messages (e.g., email/SMS messages and social media contents) or user access to a phishing website. In contrast, the detection approaches in the preparation phase, i.e., domain-registration- and certificate-based methods can find phishing websites in the early phase in which we cannot make use of any phishing messages nor user accesses for detecting phishing attacks. However, the domain-registration-based approach is limited in that not all FQDNs can be found by this approach because WHOIS records contain only domain names (i.e., websites using DDNS or hosting services cannot be found through this approach). In contrast, the key advantages of leveraging TLS certificates are (1) ability to thoroughly monitor all FQDNs of HTTPSified websites through the issued certificates even if the website owners use DDNS or hosting services, which a domain-registration-based approach may miss, and (2) ability to detect phishing websites before they are published online; these advantages are missed in the conventional crawl- and query-based approaches.

Therefore, in this paper, we address the following research question:

RQ: *How can we make use of the footprints of TLS certificates to defend against phishing attacks?*

Before answering this question, we overview the existing works that attempted to detect phishing websites by using the information contained in the TLS certificates [10]–[12]. Among them, the most recent study by Drury and Meyer [12] concluded that distinguishing malicious websites from those that are benign is difficult if they are issued certificates from the same CAs. This is because in the case that both types of websites use certificates issued by common free CAs, such as Let's Encrypt and cPanel, the certificate would have many shared fields, thus complicating their distinction.

To overcome this limitation and address the aforementioned RQ, we focused on the CN, which is the field an attacker can arbitrarily change, and the bulk registration during the survey period [1]. Several previous studies have shown that many attackers generate similar domain names in a short time [14]. In this paper, our extensive analysis of the TLS certificates corresponding to the phishing websites from CT log servers reveals that a *template*, which is a regular expression of CN obtained by analyzing the characteristics of certificates believed to have been generated by the same attacker, can be used for the following promising security applications:

- Discovering previously *unknown* phishing websites with low false positives.
- Understanding the infrastructure used to generate the phishing websites.

As shown later, our analyses reveal the existence of the phishing-website-generation service with many advanced features, such as a mass mailer to send a huge volume of customizable phishing emails, a notification mechanism, logging, analytics, and dedicated "marketplace," where customers can buy and even sell the stolen credentials. In this paper, we discuss a possible solution against such undesirable use of free service and provide a recommendation to CAs. Furthermore, we present that 24.8% of the detected phishing attacks utilize DDNS or hosting services, and 88.7% use domain names that are not listed on WHOIS database. Therefore, we can expect that our approach outperforms previous phishing-detection approaches in terms of increasing detection coverage and early detection. Note that while our approach does not aim to replace the previous defense mechanisms against phishing attacks, our experimental results indicate that our approach is an appealing complement to the conventional countermeasures against threats of phishing attacks.

The remainder of the paper is organized as follows: In Section 2, we provide a background on phishing-detection and TLS certificates. In Section 3, we present our framework that attempts to discover previously *unknown* phishing websites. Section 4 describes the methods and data used in this work. Section 5 demonstrates the statistical result of discovered phishing websites. We also highlight a case study that reveals the infrastructure used for generating groups of phishing websites. In Section 6, we provide a recommendation to CAs as well as the limitations of this study. In Section 7, we review related

1. Note that many of recent HTTPS client implementations use not only the CN field but also the subject-alternative-name (SAN) field when verifying a TLS certificate; a SAN field may contain multiple hostnames associated with the certificate [13]. We empirically found that in practice, the analysis using the CN field did not differ from that using the SAN field. We will discuss the analysis of SAN in a future study.

works and compare our results against theirs. We conclude the paper with Section 8.

2. Background: Phishing and Monitoring

Phishing is one of the most widespread cyber threats. Despite its relatively simple attack vector, the damage caused by phishing attacks is significant. The Internet Crime Complaint Center (IC3) reported that the number of victims of phishing attacks including web phishing, vishing (voice), and smishing (SMS) amounted to 26,379 in 2018, with the damage reaching 48.2 M USD [15]. Such attacks attempt to obtain sensitive information, such as credentials used for online banking, using a spoofed email address and/or a fake website that looks like an authentic one.

To mitigate the threats caused by phishing attacks, several studies have attempted to make use of features that can characterize such attacks (e.g., domain name [16], URL [17], content [18], and email address [19]). However, these approaches have several intrinsic limitations. By monitoring the registration of new domain names, a defender can proactively detect the domain names that are likely used for phishing in the future. However, as some phishing attacks leverage DDNS or hosting service with specific suffix domain names [20], [21], the approach of monitoring newly registered domain names extracted from WHOIS records will miss those cases because they contain only domain names. Similarly, a method analyzing WHOIS records, which attempts to extract domain names with the same contact information listed on the blacklist, cannot detect attacks that use DDNS or hosting services because the granularity of the analysis comprises domain names, not FQDNs. We also note the GDPR has made it infeasible to use WHOIS information because majority of WHOIS gateways have started masking information such as contact information for privacy reasons. Finally, while the phishing detection methods that leverage URLs, web content, or email messages are expected to achieve high detection accuracy [22], most of them are reactive in nature, that is, these approaches cannot detect all attacks in advance.

In contrast to the aforementioned approaches, our approach aims to proactively detect phishing websites by identifying certificates that are likely used for phishing even when the attackers utilize DDNS or hosting services in which a hostname is generated on the existing domain name. The key idea of our approach is to leverage CT logs [23]. CT is a standardized framework that aims to publish the public logs of all the issued TLS certificates. According to [24], the Chromium project started requiring all public TLS certificates issued to support CT since April 2018. Using the Censys dataset [25], we examine the certificates published after April 2018. Of the 635.7 M certificates, 99.3% of them are issued by CAs that have adopted the CT log mechanism. These CAs include freely available popular CAs such as Let's Encrypt [6] and cPanel [8], implying that all the certificates of the customers using the free certificate service are automatically registered to the public CT log servers. The CT provides the way to monitor and audit the TLS certificates issued by the publicly trusted CAs for everyone and enables defenders to efficiently identify mistakenly or maliciously issued certificates.

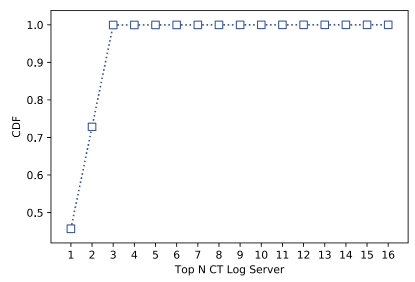


Figure 2. CDF of the number of certificates registered on Top-N CT log Servers. CT log servers are sorted according to certificate record in descending orders.

In this paper, we compiled certificate data obtained from multiple CT log servers. Newly issued certificates should be registered on one or more of the various available CT log servers, such as *argon* operated by Google and *Nimbus* operated by Cloudflare [26]. As CAs arbitrarily choose CT log servers on which to register the newly issued certificate, we need to collect certificate data from multiple CT log servers. We examined CT log servers on which certificates issued by Let's Encrypt or cPanel, both of which tend to be widely used for phishing websites, during the survey period were registered. We found that the certificates were stored on 16 CT log servers. Fig. 2 presents CDF of the number of unique certificates registered in Top-N CT log servers. As shown in the figure, when we make use of the data collected from the top CT log server, the coverage is moderate, i.e., 45.6%. However, if we use data collected from the top-3 CT log servers, the coverage becomes 99.9%. As Censys collects certificate data from a number of CT log servers, including the Top-3 servers, we used this database in our study.

3. Framework

In this section, we present our framework for discovering phishing websites. We first provide a high-level overview of the individual methodologies used in our framework. Second, we present the clustering analysis for extracting common characteristics of certificates issued for the phishing websites. Third, we describe a way to extract the intrinsic templates from the clusters. The templates can be used to discover phishing websites that have been *unknown* to the security analysts. Finally, we present a method to evaluate the effectiveness of our framework.

3.1. High-level Overview

Figure 3 illustrates a high-level overview of our framework that aims at discovering phishing websites. By analyzing the list of URLs used for phishing attacks, we first collect the TLS certificates from the corresponding websites. Next, we apply the clustering analysis to the certificates and find the group of certificates with similar characteristics (Section 3.2). We subsequently extract the intrinsic templates from the grouped certificates (Section 3.3). By applying the extracted templates to the TLS

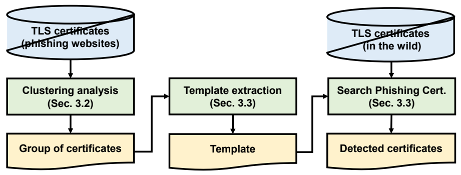

TLS certificates (phishing websites)
TLS certificates (in the wild)
Clustering analysis (Sec. 3.2)
Template extraction (Sec. 3.3)
Search Phishing Cert. (Sec. 3.3)
Group of certificates
Template
Detected certificates

Figure 3. Overview of the framework.

certificates collected from free CAs, we can discover the certificates that are likely associated with phishing attacks. Finally, we present a method to evaluate the effectiveness of our framework by using a third party tool (Section 3.4).

3.2. Grouping the Phishing Websites Using Their Certificates

We identify groups of phishing websites that are likely associated with each other. By extracting patterns that are intrinsic among each group, we expect to identify useful characteristics toward discovering phishing websites. To this end, we apply the clustering analysis to the CNs recorded in the certificates.

Before performing clustering analysis, we apply the following data preprocessing. First, we eliminate the substring "`www.`" and top-level domain names (TLDs) such as "`.com`" or "`.io`" from the CN strings, because these substrings are commonly used for all the certificates. Second, after performing filtration, we eliminate the certificates whose CNs are short. The reason for eliminating short CNs is to avoid ambiguities in determining the similarity; for instance, for a CN of short length, such as `apps(.com)`, we will detect many similar CNs, such as `apple(.com)` or `apes(.com)`. However, these CNs clearly exhibit different semantics, indicating that they are independent domain names. In this work, we empirically derive the threshold as 10.

As a clustering algorithm, we adopt DBSCAN, which enables us to eliminate certificates that are likely attributed to an attacker who does not belong to any of the existing phishing groups. As a function to measure the distance of two given strings (CNs), we leverage *Ratcliff-Obershelp* similarity [27], which is derived by recursively computing the longest common substring (LCS); for two strings "ABC" and "ADBC," the LCS is "BC." First, we find the LCS of the two given strings. We subsequently split each string using the detected LCS as a separator and attempt to find an LCS again for the pairs of strings at both sides. This operation is performed recursively until there are no characters in common between the split strings. *Ratcliff-Obershelp* similarity for two strings (x, y) is defined as $d(x, y) = 2M/T$, where M is the sum of the lengths of LCSs obtained in the above operation between x and y, and $T = |x|+|y|$, where $|s|$ denotes the length of a string s. This similarity is expressed as a ratio between 0 and 1, and then used as a normalized distance for DBSCAN.

Finally, we require useful heuristics to analyze CNs, which have variable lengths and domain name structures. Thus, we introduce a variable m, which denotes the number of dots in a given CN. For instance, $m = 1$ for `ieee.org` and $m = 3$ for `www.cs.example.edu`.

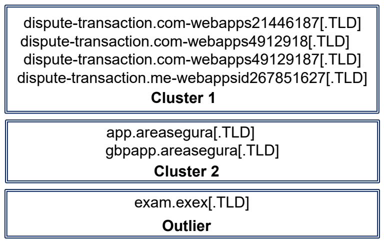

dispute-transaction.com-webapps21446187[.TLD]
dispute-transaction.com-webapps4912918[.TLD]
dispute-transaction.com-webapps49129187[.TLD]
dispute-transaction.me-webappsid267851627[.TLD]
Cluster 1
app.areasegura[.TLD]
gbpapp.areasegura[.TLD]
Cluster 2
exam.exex[.TLD]
Outlier

Figure 4. An example of the clustering result.

TABLE 1. An example of clustering results for CNs with $m = 2$. $\epsilon = 0.25$, 0.30, and 0.35.

	Cluster
$\epsilon = 0.25$	`login.portaleprivatimps[.TLD]`
	`secure.portaleprivatimps[.TLD]`
	`accesso.portaleprivatimps[.TLD]`
$\epsilon = 0.30$	`login.portaleprivatimps[.TLD]`
	`secure.portaleprivatimps[.TLD]`
	`accesso.portaleprivatimps[.TLD]`
	`secure.mpsprivati[.]com`
$\epsilon = 0.35$	`login.portaleprivatimps[.TLD]`
	`secure.portaleprivatimps[.TLD]`
	`accesso.portaleprivatimps[.TLD]`
	`secure.mpsprivati[.TLD]`
	`certificazione.areaprivatimps[.TLD]`
	`certificazione.portalemps[.TLD]`
	`certificazione.mpsprivati[.TLD]`

The insight behind these heuristics is that CNs generated by the same attacker are expected to use a fixed domain name structure, implying that the number of dots used for these CNs should be the same.

Figure 4 demonstrates an example of the clustering result. If several certificates have similar CNs, we group them as a cluster. If some certificates have CNs dissimilar to any of those in the found clusters, we eliminate such certificates as outliers. DBSCAN has two parameters. We adjust the first parameter ϵ, which controls the similarity between the CNs in a cluster. We set another parameter $minPts$, which is the minimum number of certificates in a cluster, as $minPts = 2$.

Table 1 presents an example of clustering results with different values of ϵ. Here, we select a case in which the number of dots is set to $m = 2$. As shown in the figure, when ϵ is 0.25, all the three CNs in the cluster look similar. For the other cases, the CNs in a cluster contain dissimilar CNs. Thus, as illustrated through this example, we empirically adopt the parameter as $\epsilon = 0.25$ for $m = 2$. For other m, following the same procedure, we empirically derive the thresholds as 0.24 ($m = 1$), 0.3 ($m = 3$), 0.33 ($m = 4$), and 0.35 ($m \geq 5$), respectively.

3.3. Extracting Template

Figure 5 illustrates the process of extracting templates from the clusters obtained in Section 3.2. In the case of preprocessing, we eliminate the substring "`www.`'' and TLDs in a way similar to what was described in Section 3.2. First, we extract all the substrings common to CNs in a cluster if the substring is three or more characters long. This process is applied to all the strings, which are divided by a dot. Next, we convert the strings other than the common substrings of each CN in the cluster to regular expressions (regexps) and subsequently combine them. When combining regexps, the common substrings are not modified, and we combine the minimum and maximum lengths of the regular expressions. For example, combining the regexps, [a-z]{3}, [a-z]{5}, and [a-z]{4}, yields the regexp [a-z]{3,5}.

Finally, we verify the genericity of the generated regexps. If a regexp for detecting phishing websites is too generic, it will also detect other legitimate websites, thereby causing large false positives. To test the genericity of a regexp, we adopt *entropy reduction* proposed by Xie [28]. Let e be a regexp. Let $B_e(u)$ be the average number of bits (information entropy) required to encode the representation in binary when using the regexp e to represent a string u. Similarly, let $B(u)$ be the information entropy to represent a string u without using the regexp. Information entropy for a random string can be calculated as $L \log_2 N$, where L is the number of characters that constitute u. N is the number of available characters. It is well known that the information entropy defined in this way is used for measuring the strength of passwords. The information entropy of an original string u and its regexp are calculated as follows.

$$B(u) \quad = L \log_2(A + D) \tag{1}$$

$$B_e(u) = \overline{L_a} \log_2 A + \overline{L_d} \log_2 D + \overline{L_{ad}} \log_2(A + D) \tag{2}$$

where $\overline{L_a}$, $\overline{L_d}$ and $\overline{L_{ad}}$ are the average number of characters represented by the regular expressions `[a-z]` (alphabet), `[0-9]` (digits), and `[a-z0-9]` (alphabet + digits), respectively. A and D denote the number of characters that can be represented by `[a-z]` and `[0-9]`, with $A = 26$ and $D = 10$.

Next, we introduce a metrics termed as *entropy reduction*, which measures the amount by which a regexp reduces the information entropy to represent a string; i.e., entropy reduction is calculated as $d(e) = B(u) - B_e(u)$. If a regexp has a small $d(e)$, the information entropy of the regexp e is relatively large, implying the expression is generic. Using a regexp with a large entropy for detecting phishing certificates may result in several false positives owing to its high genericity. Therefore, we extract regexps with d greater than or equal to preset threshold. We empirically derived the threshold as 55. After careful manual inspection, we decided to set a heuristics to eliminate the substrings that are used for the domain names of DDNS or the hosting services. The domain names used by these services are not necessarily limited to use only for phishing websites.

Example: For the purpose of illustration, we present an example of template extraction process. Suppose that we obtain two CNs, `apple-accountverify123[.TLD]`[2] and `payment-accountverify55[.TLD]` in a cluster. The substring common to the two strings is `-accountverify`. The regexps of these CNs are `[a-z]{5}-accountverify[0-9]{3}[.TLD]` and `[a-z]{7}-accountverify[0-9]{2}[.TLD]`.

2. Throughout this study, we replace the top-level domain part with the string [.TLD] to mask the phishing URLs.

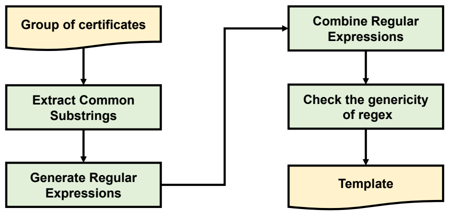

Group of certificates
Extract Common Substrings
Generate Regular Expressions
Combine Regular Expressions
Check the genericity of regex
Template

Figure 5. Process of template extraction.

Combining the two regexps yields the following regexp, `[a-z]{5,7}-accountverify[0-9]{2,3}[.TLD]`. We now calculate the *entropy reduction*. The number of characters that constitute `-accountverify` is 14, and the average number of characters of the regexp part $\overline{L_a}$ and $\overline{L_d}$ are 6 and 2.5, so the total string length L is the sum of these, 22.5. Thus, using Eqn. 1 and Eqn. 2, we obtain the following results: $B(u) = 116.3$, $B_e(u) = 36.5$, and $d(e) = 79.8$. Since the entropy reduction exceeds the threshold 55, we adopt the regexp as a template. Using this template, a certificate whose CN is `google-accountverify37[.TLD]` is detected as the one used by phishing websites. However, although `security-accountverify9[.TLD]` contains the same substring, our approach does not detect it because the number of characters of the regexp is different from those of the template.

3.4. Evaluation Approach

We present a method of evaluating the correctness of the detected phishing websites. A straightforward approach we present to evaluate the aforementioned correctness is to examine the websites we detected. To this end, several existing tools such as web client type honeypot can be utilized. However, among the detected websites, there were extremely few active websites that we could access; this is because malicious websites are usually short-lived. Therefore, we leverage VirusTotal [29], which is the most popular online virus scanner service. VirusTotal inspects a target file or URL with over 70 antivirus software and URL/domain blacklisting services.

Our approach attempts to discover potential phishing websites at the time of TLS certificate issuing phase, implying that we can detect phishing websites before they are actually used. As it may take a considerable amount of time before a domain name is posted to VirusTotal, we performed scanning of the discovered domain names after a certain time of period has passed since the collection of TLS certificates. We note that VirusTotal may have missed several phishing domain names, i.e., it should involve false negatives. Likewise, it should also include false positives. Despite these limitations, we believe that analyzing the outputs of the VirusTotal will provide us with promising means to evaluate the effectiveness of our approach at scale — detecting phishing websites at the time of TLS certificate issue.

4. Data

In this study, we leverage the following two certificate datasets: the blacklisted certificates used for creating tem-

TABLE 2. Number of phishing certificates for each m.

$m=1$	$m=2$	$m=3$	$m=4$	$m \geq 5$	Total
956	468	110	70	30	1,634

plates and certificates issued by free CAs for searching phishing websites in the wild. To collect the certificates of the phishing websites, we use the data collected from OpenPhish [30], a publicly available collection of phishing URL feeds. We note that our analysis is not limited to these data and can be applied to other blacklists such as Phishtank. We collect the phishing URLs from October 2018 to January 2019. For each URL we collect, we obtain the corresponding certificates stored at CT log servers by using the Censys database [31] (See Section 2). In total, we extract 2,638 unique certificates. After we apply the data preprocessing described in Section 3.2, we obtain 1,634 unique certificates, which were reported as having been used for the phishing websites.

Table 2 presents the number of certificates we derive for each m, which is the number of dots in a CN. We can see that the majority of certificates had CNs with a small number of dots; $m \leq 2$ for more than 87% of the certificates, while a non-negligible number of certificates had CNs with a large number of dots; $m \geq 4$ for more than 5% of the certificates. The high variability of m implies that we need to carefully adjust the thresholds for finding the certificates that have visually similar CNs.

We inspect the CAs that issued the certificates used for phishing websites and found that the majority were issued by two free CAs; 852 (50.9%) of them were issued by Let's Encrypt and 714 (42.7%) are issued by cPanel [8]. Thus, HTTPSified phishing websites can be efficiently identified by searching for the CT logs of these CAs. Given this observation, we collect the certificates issued by these two CAs. We collect 38,669,178 certificates issued by these free CAs; 54.9% of these were issued by Let's Encrypt and the remaining 45.1% by cPanel. We use these data as the basis of our analysis shown in Section 5.3, in which we aim to discover phishing websites.

5. Results

In this section, we present the results using the framework described in Section 3 and the data presented in Section 4. We first present the detected clusters of TLS certificates (Section 5.1), and the *templates* extracted from the clusters (Section 5.2). Next, we present the discovered phishing certificates using the templates (Section 5.3) and then validate them (Section 5.4). Finally, we perform an in-depth analysis of the detected phishing certificate through a case study (Section 5.5). We demonstrate that the analysis enables the learning of the infrastructure of the phishing websites.

5.1. Clusters of Certificates

Applying the DBSCAN algorithm to the CNs of the phishing websites resulted in 106 of distinct clusters. These clusters include 341 (20.8%) certificates out of 1,634, which is the number of certificates covered in this study. We note that the remaining 1,293 of certificates were not grouped into any of clusters due to the configuration of the DBSCAN algorithm, i.e., $minPts = 2$. This observation implies that there are varieties of certificates targeting various websites, using different schemes. We conjecture that by increasing the sample size of phishing websites, the clustering process will generate more clusters.

TABLE 3. Clustering results.

	$m=1$	$m=2$	$m=3$	$m=4$	$m \geq 5$	Total
#clusters	47	39	8	7	5	106
#certificates	124	122	49	33	13	341

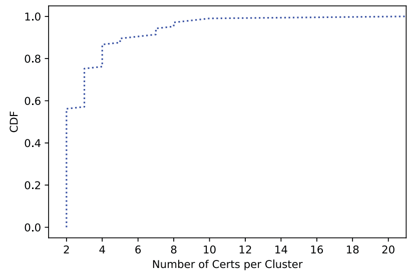


Figure 6. CDF of number of certificates per cluster.

Table 3 shows the clustering result for each m. As we have shown in Table 2, majority of the clusters and certificates were concentrated to small m, i.e., $m \leq 2$. Figure 6 presents the distribution of the number of certificates in each cluster. We see that the sizes of each cluster are small in general, while there are non-negligible number of clusters that had a large number of certificates.

5.2. Extracted Templates

Using the 106 clusters, we extracted 69 templates that had the *entropy reduction rate* greater than the pre-determined threshold presented in Section 3. Table 4 presents the examples of domains (CNs) in the two clusters and the extracted templates. We notice that several domains shown in the aforementioned table include those provided by DDNS or hosting services; e.g., `serveirc[.TLD]` and `hoster-test[.TLD]`. The observation shows evidence that attackers leverage DDNS and/or hosting services as the infrastructure of the phishing websites. We found that 10 (14.5%) of the extracted templates contained such domains.

Furthermore, these results suggest some phishing attackers tend to put deceptive strings (for example, "verify-web" and "onedrive" in the table) into all the FQDNs to trick users into believing that the websites are legitimate if they perform similar phishing attacks several times.

5.3. Discovered Phishing Certificates

Using the method described in Section 3, we search for the certificates of the websites that are likely used for phishing attacks. Of the 38.7 M of certificates collected from Let's Encrypt and cPanel, we identified 1,650 certificates that are considered to have been used for phishing. Notably, all the detected certificates had *not* been listed on the OpenPhish blacklist, implying that they were unknown

TABLE 4. EXAMPLES OF CNS IN CLUSTERS AND THE EXTRACTED TEMPLATES.

	Cluster $(m = 2)$	Cluster $(m = 4)$
CN	`verify-webapps25476.serveirc[.TLD]`	`onedrive.liveviewuserauthaspx209hr28jh.srv156794.hoster-test[.TLD]`
	`verify-webapps72647.serveirc[.TLD]`	`onedrive.liveviewuserauthaspx209hr28jh.srv156816.hoster-test[.TLD]`
	`verify-webscrid2678.serveirc[.TLD]`	`onedrive.liveviewuserauthaspx209hr28jh.srv156797.hoster-test[.TLD]`
		`onedrive.liveviewuserauthaspx209hr28jh.srv156796.hoster-test[.TLD]`
Template	`verify-web[a-z0-9]{4,5}.serveirc[.TLD]`	`onedrive.liveviewuserauthaspx209hr28jh.srv156[0-9]{3,3}.hoster-test[.TLD]`

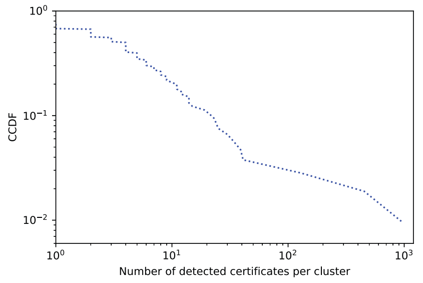


Figure 7. Log-log CCDF of the number of detected certificates per cluster.

at the phase of certificate issuance. Figure 7 presents the log-log complementary cumulative distribution (CCDF) of the number of detected certificates per cluster. We see that the distribution is *heavy-tailed*; while majority of clusters had a small number of or even zero similar certificates, there are non-negligible number of clusters that had a large number of previously unknown certificates. Specifically, the top-2 clusters had 924 and 395 of the discovered certificates. The existence of clusters with the large number of similar certificates indicates that they likely automate the process of generating the phishing websites.

The cluster with 395 certificates showed that all made use of hosting services because all the CNs had the domain name suffix of `zap-hosting[.TLD]`. We also detected 15 additional certificates with CNs provided by other DDNS or hosting services; therefore, 410 (24.8%) of the detected attacks used DDNS or hosting services. Furthermore, 88.7% of the CNs of the discovered certificates did not meet the criteria to be listed on WHOIS, i.e., the CNs have one or more labels in addition to effective TLD. Conventional domain-registration-based approaches fail in detecting such attacks as WHOIS records do not contain all FQDNs. In Section 5.5, we perform an in-depth analysis using the cluster containing 924 of certificates.

5.4. Evaluation

We obtained 1,049 unique CNs after eliminating the duplicated CNs; we obtained 1,049 candidates of the phishing websites. Note that some of the discovered certificates had the same CNs because they were issued several times during the survey period. First, the verification with VirusTotal revealed that for 90.8% of the websites we detected, at least one antivirus checker raised alarms while for 72.5% of the websites we detected, at least two antivirus checkers raised alarms. As mentioned in Section 6.1, the possibility remains that the antivirus checkers in VirusTotal overlook malicious ones because malicious websites are usually short-lived. Therefore, some of 9.2% of CNs may include potentially malicious ones, and we cannot determine that they are false positives of our approach. The aforementioned results clearly demonstrate that our framework was able to find a lot of potentially malicious websites and the majority of them were identified as obviously malicious by third-parties. For example, a template `allegro.pl-login.form-a[a-z0-9]{0,12}.[a-z0-9]{0,6}[.TLD]` detected 10 websites, eight of which were detected as malicious with VirusTotal. Through a careful manual inspection of the corresponding certificates, we conjecture that the other two were also generated by the same attacker. At least, we were unable to identify evidence that the remaining two CNs were used for legitimate services.

However, given actual security operations, our approach should be used to account for potential false positives. A practical usage is a pre-filter to extract highly suspicious ones from large certificates and send them for manual inspection.

5.5. In-Depth Analysis

We present an in-depth analysis of the detected phishing certificates using a template that yielded the largest number of phishing websites. The template is `[a-z]{6,8}.runescape.com-[a-z]{1,8}[.TLD]`. Using this template, we detected the following two patterns of domain names:
`secure.runescape.com-[a-z]{1,8}[.TLD]`
and
`services.runescape.com-[a-z]{1,8}[.TLD]`.
A simple domain name analysis revealed that these domain names target *Runescape*, which is a massively popular multiplayer online role-playing game (MMORPG). We demonstrate examples of CNs of the certificates in a cluster targeting *Runescape* in the left column of Table 5. We note that 863 (93.4%) out of 924 certificates were issued by the same CA, Let's Encrypt. We also note that in this case, the combinations of TLDs and second-level domains are often different.

Our manual inspection on the discovered certificates revealed that these certificates are generated by a phishing website generation service, which is sold by a rogue company. Although searching the web will reveal such companies in the wild, we refrain from specifying the name of company for the ethical reason. In order to confirm whether the service actually generates certificates with CNs that match the templates we identified, we subscribe to their service to check the certificates in the service. We note that we do not use any of the services provided by the company. As shown in the right column of Table 5, the certificates generated by the kit match to the templates we constructed.

In addition, we found that the phishing website generation service provides many advanced features to help an attacker perform the phishing attack efficiently; e.g., mass mailer to send a huge volume of customizable phishing email, notification mechanism, logging, analytics, and

TABLE 5. EXAMPLES OF CNS FOR THE DISCOVERED CERTIFICATES FOR A CLUSTER AND USED IN A PHISHING WEBSITE GENERATION KIT TARGETING *Runescape*.

cluster	phishing kit
`services.runescape.com-an[.TLD]`	`services.runescape.com-rv[.TLD]`
`secure.runescape.com-mq[.TLD]`	`secure.runescape.com-ao[.TLD]`
`secure.runescape.com-g[.TLD]`	`secure.runescape.com-vo[.TLD]`
`secure.runescape.com-l[.TLD]`	`secure.runescape.com-rs[.TLD]`

dedicated "marketplace" where customers can buy and/or even sell the stolen credentials. We note that although previous studies [32], [33] have mentioned the existence of the phishing website generation service, these studies did not provide the deep insight into the ecosystem of the service. Given these results and observations, we may conclude that the analysis of certificates can reveal the infrastructure and ecosystem of the phishing attack.

6. Discussion

In this section, we discuss the limitations of this work and the undesirable use of free services such as free CAs, and provide a recommendation to the CAs.

6.1. Limitations

Bias inherent in our dataset.

As our analysis relies on the URLs collected from the OpenPhish dataset, it is possible that our choice introduces some bias. In the future, we plan to replicate our work using other phishing URLs.

Evaluating the phishing websites

While we systematically extracted certificates that are likely associated with the existing phishing websites, we did not have direct access to those discovered websites, because phishing websites are short-lived in nature. Therefore, we were unable to confirm whether the websites marked as malicious by VirusTotal actually performed phishing activities. Additionally, as we mentioned earlier, VirusTotal is not a perfect solution, i.e., it involves both false negatives and false positives. To verify the activities of detected websites, it is crucial that we have real-time monitoring/analysis system so that we can check the activities of the possible phishing websites in a timely manner. Building such a monitoring system therefore remains a future work.

Threats to validity.

While the majority of the websites discovered by our method were flagged by VirusTotal, there were other several websites not flagged. Careful manual inspection revealed that among the undetected websites, there were a few false positives in our approach. If a template contains universal words such as `service` and `communication`, which are often used for benign websites, it becomes difficult to distinguish between a benign website and a malicious website. Here is an example. While template `officespace{1,2}[a-z][.TLD]` detected 31 websites, 12 of them were flagged by VirusTotal. Since the words "office" and "space" are both frequently used, the template detected several benign as well as malicious websites. A promising method of preventing this phenomenon is to collect and list universal words in advance, and reduce the value of the *entropy reduction* accordingly if such universal words are included in the template. This decreases the number of false positives because such highly generic templates with a low value of the *entropy reduction* will be eliminated.

Another false-positive case may occur under the following two conditions: (1) several legitimate websites with similar domain names are mistakenly included in the blacklist and the template is created for them, and (2) certificates of other legitimate websites with domain names to be matched by the template are issued. We note that these events rarely occur and we did not find this case in this work.

Wild card certificate.

Even if an attacker generates multiple CNs and performs phishing, it is difficult to investigate them using our method if the attacker uses wild card certificates. However, using such certificates can be a disadvantage for attackers because if one of the hosts they use is blacklisted, the other hosts will probably be disabled by antivirus software, Google Safe Browsing, etc. Hence, if attackers intend to generate many similar CNs and perform phishing, they would benefit by changing the domain part (as does the phishing kit discovered in this study) and issue certificates accordingly.

6.2. Detection Evasion

An attacker could efficiently perform phishing attacks under the constraints on time and strings that are effective for creating phishy URLs. For the time constraint, a large amount of the certificates for a phishing website are issued for a short period. For the string constraint, the URL of phishing website must include deceptive strings to make victims believe that the prepared website is genuine. The examples of deceptive strings are specific brand names, generic terms (service, account, etc.), and actions (login, pay, registration, etc.). Our analysis works on the basis of these attacker constraints. The attacker ignoring the above-mentioned constraints may fail to deliver efficient phishing attacks (e.g., issuing certificates over a long period and using fully randomized domain names). This is why we especially focused on phishing websites among malicious activities. A possible method, especially against a string constraint, to complicate our analysis is to use "leetspeak."

Suppose that there is a benign website with the domain name `login-account-service[.TLD]`, and an attacker tries to impersonate the website and issues three certificates, the CNs of which are `login-account-serv1ce[.TLD]`, `l0g1n-acc0unt-serv1c3[.TLD]`, and `l0gin-4ccount-s3rv1ce[.TLD]`. In this case, the following two problems may occur: (1) the created template is too generic to use and (2) we detect the benign website by the template. As for the first problem, even if an attacker uses leetspeak, those certificates will surely be incorporated into the same cluster by DBSCAN clustering because of their high degrees of similarity; the average *Ratcliff–Obershelp* similarity between them is 0.778. However, the template we obtained from these CNs (`[a-z0-9]{10}unt-s[a-z0-9]{6}[.TLD]`) after applying the proposed method is too generic for detecting phishing sites as the value of the entropy reduction is 36.19, which is much lower than the preset threshold. As a countermeasure, we can decode leetspeak by using a tool, such as Universal Leet Converter [34], during template extraction and phishing detection. In this example, we obtain `login-account-service[.TLD]` as a template by decoding the leetspeaks. On the other hand, this template generation may create false positives because this generated template simply matches the legitimate one. To eliminate such false positives, we can use such template for matching only certificates with CN including leetspeaks. Incorporating these improvements into the detection system is left for future work.

6.3. Recommendations

The number of phishing websites with HTTPS have been increasing. Some countermeasures are essential considering that most use free certificates issued by Let's Encrypt or cPanel. Sectigo Ltd. [35], which operates cPanel, specifies in its certificate practice statement (CPS) that if a certificate is found to have been used for illegal purposes, such as phishing and malware, they will revoke it within 7 days [36]. On the other hand, Let's Encrypt terminated efforts to confirm websites were not malicious using Google Safe Browsing, because they consider that domain validation (DV) certificates are only intended to secure communications between the client and server, not to ensure the safety of the website. However, as shown in Section 5.5, 93.4% of the certificates of *similar CNs* that are considered to be generated by the phishing kit targeting *Runescape* were issued by Let's Encrypt, and CNs with those specific patterns can be found easily using our method. Considering these findings, the CA should identify such CNs using the approach presented in this study and revoke the certificates.

7. Related Work

In this section, we review related works and discuss the comparison between them and our research.

7.1. Detecting URLs and Contents.

Multiple studies have shown that features extracted from URLs and content can be used as clear indicators to detect phishing websites. The features are created through the following expert knowledge: lexical anomalies in URLs (e.g., blacklisted words, hyphens used instead of dots, and brand/service names in the URL path) [37], [38], IP address used as the domain name in the URL [38], [39], many dots in the URL [37], [39], inconsistent brand names/logos (e.g., the *brand-X* name not on a *brand-X* domain name) [40], similarity among contents [41], [42], and so on. CANTINA and CANTINA+ are complementary approaches using the above heuristics. They examine the content to determine whether the website is legitimate or not by using search results of important terms in the content extracted by the term frequency-inverse document frequency (TF-IDF) algorithm [18], [43]. These approaches based on URL and content features successfully detect phishing websites. However, their limitation is that they detect only visited websites or listed websites (e.g., the URLs in delivered emails). In other words, detecting never-accessed/-listed websites is beyond the scope of these approaches. Our method does not face such limitations, because it relies on the certificates that anyone can comprehensively list via the certificate transparency (CT) log server [23] or a repository of Internet scannings such as Censys [25].

7.2. Detecting Certificates.

Given the rapid increase in the number of HTTPSified phishing websites, there have been some attempts to detect phishing websites using the certificates [10]–[12]. Torroledo et al. [10] and Dong et al. [11] proposed methods for identifying malicious use of certificates based on the features included in the fields of the certificate. However, Drury and Meyer mentioned some fields are very similar (or the same) for all certificates issued by the same issuer, and concluded that it is generally difficult to differentiate certificates of phishing websites from those of benign websites if the certificates of both phishing and benign websites are provided by the same issuer [12]. While previous studies make use of features from the certificates to identify differences between the certificates of benign and phishing websites, our study reveals that we can make valued use of the information obtained from the certificates; that is, we can discover previously unknown phishing websites, systematically find targeted websites, and understand the infrastructure used for generating such phishing websites.

7.3. Phishing Kit and Evasion.

Criminals create phishing kits, which are packages used to deploy a phishing website on a web server. They sell phishing kits in underground marketplaces and accept custom requests for kit creation [33], [44]. Phishing kits include server-side and client-side evasion techniques using server directives (`.htaccess` files), server-side scripts, and JavaScript to interfere with detection by the security community [33], [45]. The evasion is carried out based on a client IP address, referrer, and user agent. If the accessing client environment is detected by evasion techniques, the content would not be available. We emphasize that in most cases, our analysis is not affected by such evasion techniques because our approach leverages the characteristics of TLS certificates, which can be collected from the publicly available CT logs.

8. Conclusion

This work focuses on the fact that phishing websites have started adopting HTTPS; this could expose their intrinsic features that could be used to detect them in a systematic manner. Compared to conventional phishing-detection approaches, certificate-based approach has the following advantages: it allows a defender (1) to comprehensively monitor all HTTPSified websites through the issued certificates, even if the attacker utilizes DDNS or hosting services, and (2) to detect phishing websites before they are published on the Internet.

Although some previous studies have reported that distinguishing a benign website from a malicious website by using certificate information alone is difficult, we established a framework to discover unknown phishing websites through a *template* extracted using attackers' bulk registration and CN in certificates. We demonstrated that the template can be applied not only to discover phishing websites with low false positives but also to understand the infrastructure used to generate the phishing websites, e.g., phishing-website-generation kit. We also demonstrated that our proposed approach can find several types of phishing websites that existing approaches cannot detect in nature because these websites make use of DDNS or hosting services, which are not listed in domain name-based database such as WHIOS. We believe that our approach contributes to complement the lack of the various existing phishing-detection techniques and sheds new light on the *footprints* of TLS certificates as a key to understanding the origin of threats.

References

[1] Google, "Https encryption on the web," https://transparencyreport.google.com/https/overview?hl=en.

[2] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring HTTPS adoption on the web," in *26th USENIX Security Symposium*, 2017.

[3] Google, "Webmaster central blog, https as a ranking signal," https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html.

[4] ——, "A secure web is here to stay," https://security.googleblog.com/2018/02/a-secure-web-is-here-to-stay.html.

[5] Mozilla, "Communicating the dangers of non-secure http," https://blog.mozilla.org/security/2017/01/20/communicating-the-dangers-of-non-secure-http/.

[6] Let's Encrypt, https://letsencrypt.org/.

[7] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafò, K. Papagiannaki, and P. Steenkiste, "the cost of the "s" in https."

[8] cPanel, https://cpanel.net/.

[9] APWG, "Phishing activity trends report 3rd quarter 2019," https://docs.apwg.org/reports/apwg_trends_report_q4_2019.pdf.

[10] I. Torroledo, L. D. Camacho, and A. C. Bahnsen, "Hunting malicious tls certificates with deep neural networks," in *Proc. of ACM AIsec*, October 2018.

[11] Z. Dong, A. Kapadia, J. Blythe, and L. J. Camp, "Beyond the lock icon: Real-time detection of phishing websites using public key certificates," in *Proc. of APWG Symposium eCrime*, 2015.

[12] V. Drury and U. Meyer, "Certified phishing: Taking a look at public key certificates of phishing websites," in *Proc. of USENIX Symposium SOUPS*, 2019.

[13] S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell, and D. Cooper, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, May 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5280.txt

[14] X. Li, G. Geng, Z. Yan, Y. Chen, and X. Lee, "Phishing detection based on newly registered domains," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 3685–3692.

[15] Internet Crime Complaint Center, "2018 internet crime report," https://pdf.ic3.gov/2018_IC3Report.pdf.

[16] H. Shirazi, B. Bezawada, and I. Ray, "Kn0w thy doma1n name": Unbiased phishing detection using domain name based features," in *Proc. of ACM SACMAT*.

[17] A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical feature based phishing url detection using online learning," in *Proc. of ACM AISec*. ACM, 2010, pp. 54–60.

[18] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, 2007.

[19] A. van der Heijden and L. Allodi, "Cognitive triaging of phishing attacks," in *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019.

[20] P. Peng, C. Xu, L. Quinn, H. Hu, B. Viswanath, and G. Wang, "What happens after you leak your password: Understanding credential sharing on phishing sites," in *Proc. of ACM CCS*, 2019.

[21] PHISHLABS, "2019 phishing trends and intelligence report the growing social engineering threat," https://info.phishlabs.com/hubfs/2019PTIReport/2019PhishingTrendsandIntelligenceReport.pdf.

[22] L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. Nguyen, "A novel approach for phishing detection using url-based heuristic," in *2014 International Conference on Computing, Management and Telecommunications (ComManTel)*, April 2014, pp. 298–303.

[23] Google, "Certificate transparency," https://www.certificate-transparency.org.

[24] The Chromium Project, "Chromium Certificate Transparency Policy," https://github.com/chromium/ct-policy.

[25] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by Internet-wide scanning," in *ACM CCS*, 2015.

[26] B. Laurie, A. Langley, and E. Kasper, "Certificate transparency," Internet Requests for Comments, RFC Editor, RFC 6962, June 2013.

[27] J. W. Ratcliff and D. E. Metzener, "Pattern-matching-the gestalt approach," *Dr Dobbs Journal*, vol. 13, no. 7, p. 46, 1988.

[28] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming botnets: Signatures and characteristics," *ACM SIGCOMM CCR*, vol. 38, pp. 171–182, 01 2008.

[29] VirusTotal, "Virustotal," https://www.virustotal.com/.

[30] OpenPhish, "Openphish faq," https://openphish.com/faq.html.

[31] "Censys," https://censys.io/.

[32] P. Peng, C. Xu, L. Quinn, H. Hu, B. Viswanath, and G. Wang, "What happens after you leak your password: Understanding credential sharing on phishing sites," 07 2019, pp. 181–192.

[33] A. Oest, Y. Safei, A. Doupé, G.-J. Ahn, B. Wardman, and G. Warner, "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in *APWG Symposium on Electronic Crime Research (eCrime)*, 2018, pp. 1–12.

[34] Robert Ecker, "Universal leet (l337, l33t, 1337) converter," http://www.robertecker.com/hp/research/leet-converter.php.

[35] Sectigo, https://sectigo.com/.

[36] ——, "Sectigo certification practice statement (version 5.1.1)," https://sectigo.com/uploads/files/Sectigo-CPS-v5.1.1.pdf.

[37] A. Le, A. Markopoulou, and M. Faloutsos, "Phishdef: Url names say it all," in *Proceedings of 2011 IEEE INFOCOM*, 2011.

[38] R. Verma and K. Dyer, "On the character of phishing urls: Accurate and robust statistical learning classifiers," in *Proc. of ACM CODASPY 2015*.

[39] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Procc of International Conference WWW*, 2007.

[40] S. Marchal, K. Saari, N. Singh, and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets," in *Proc. of IEEE ICDCS 2016*.

[41] J. Mao, W. Tian, P. Li, T. Wei, and Z. Liang, "Phishing-alarm: Robust and efficient phishing detection via page component similarity," *IEEE Access*, vol. 5, pp. 17 020–17 030, 2017.

[42] 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 *Proc. of ESORICS*, 2017.

[43] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing web sites," *ACM TISSEC*, 2011.

[44] D. Birk, S. Gajek, F. Grobert, and A.-R. Sadeghi, "Phishing phishers - observing and tracing organized cybercrime," in *Proc. of ICIMP '07*.

[45] A. Oest, Y. Safaei, A. Doupé, G. Ahn, B. Wardman, and K. Tyers, "Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists," in *2019 IEEE SP*.