# Security Analysis on Practices of Certificate Authorities in the HTTPS Phishing Ecosystem

Doowon Kim[1], Haehyun Cho[2], Yonghwi Kwon[3],
Adam Doupé[4], Sooel Son[5], Gail-Joon Ahn[4,6], Tudor Dumitras[7]
[1]University of Tennessee, Knoxville [2]Soongsil University [3]University of Virginia
[4]Arizona State University [5]KAIST [6]Samsung Research [7]University of Maryland, College Park
doowon@utk.edu, haehyun@ssu.ac.kr, yongkwon@virginia.edu,
doupe@asu.edu, sl.son@kaist.ac.kr, gahn@asu.edu, tudor@umd.edu

## ABSTRACT

Phishing attacks are causing substantial damage albeit extensive effort in academia and industry. Recently, a large volume of phishing attacks transit toward adopting HTTPS, leveraging TLS certificates issued from Certificate Authorities (CAs), to make the attacks more effective. In this paper, we present a comprehensive study on the security practices of CAs in the HTTPS phishing ecosystem. We focus on the CAs, critical actors under-studied in previous literature, to better understand the importance of the security practices of CAs and thwart the proliferating HTTPS phishing. In particular, we first present the current landscape and effectiveness of HTTPS phishing attacks comparing to traditional HTTP ones. Then, we conduct an empirical experiment on the CAs' security practices in terms of the issuance and revocation of the certificates. Our findings highlight serious conflicts between the expected security practices of CAs and reality, raising significant security concerns. We further validate our findings using a longitudinal dataset of abusive certificates used for real phishing attacks in the wild. We confirm that the security concerns of CAs prevail in the wild and these concerns can be one of the main contributors to the recent surge of HTTPS phishing attacks.

## CCS CONCEPTS

• **Security and privacy → Web application security**.

## KEYWORDS

phishing attacks; PKI; TLS; certificates; CA

**ACM Reference Format:**
Doowon Kim, Haehyun Cho, Yonghwi Kwon, Adam Doupé, Sooel Son, Gail-Joon Ahn, Todor Dumitras. 2021. Security Analysis on Practices of Certificate Authorities in the HTTPS Phishing Ecosystem. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (ASIA CCS '21), June 7–11, 2021, Virtual Event, Hong Kong.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3433210.3453100


*ASIA CCS '21, June 7–11, 2021, Virtual Event, Hong Kong*

ACM ISBN 978-1-4503-8287-8/21/06…$15.00
https://doi.org/10.1145/3433210.3453100

## 1 INTRODUCTION

Phishing attacks are a critical security threat to millions of Internet users. The victims of online criminals sustain tremendous financial losses—more than $3.5 billion in 2019, according to the US Federal Bureau of Investigation (FBI) [49]. The volume of the attacks is continuously growing, and the attacks have become *more* prevalent online threats than malware websites [33, 61].

Particularly, recently the *HTTPS* phishing attacks have radically surged and have taken the place of HTTP phishing attacks [23]. In this attack, adversaries exploit users' misunderstanding of the security indicators (*e.g.*, padlock icon) in modern web browsers [18, 19, 44]. The indicators are rendered only when the confidentiality and integrity of their communications as well as the authenticity of a website are established. However, users tend to incorrectly perceive the meaning of the security indicators, and often mistakenly believe that phishing websites are legitimate, benign, and even trustful [38, 43, 52, 55]. These misunderstandings can significantly increase the likelihood for users to be tricked by HTTPS phishing websites [38, 43, 52, 55].

HTTPS phishing attacks can be successfully launched (*i.e.*, properly displaying security indicators on web browsers) only when valid certificates are issued and served with the attacks. HTTPS relies on the Public Key Infrastructure (called the Web PKI) where only publicly trusted third-parties, called Certificate Authorities (CAs), can issue certificates after verifying the ownership of domains. As CAs are responsible for the issuance and management of certificates, they play a critical role in the Web PKI, and they are directly involved in the HTTPS phishing ecosystem.

Anecdotes demonstrate how critical CAs are in the HTTPS ecosystem. Particularly, in 2011, a Dutch CA was actually compromised, resulting in the mis-issuance of certificates. Subsequently, the mis-issued ones were used for a man-in-the-middle attack against Google in Iran [11]. We believe that better understanding and analyzing the security practices of CAs from the point of attackers' view are the first step to mitigate the HTTPS-related attacks including HTTPS phishing attacks. However, the security practices have not been systematically measured and consequently not fully understood. Prior work [34, 36, 46, 51, 56] has mainly focused on understanding phishing attackers, not CAs. Particularly, they aimed to fully identify phishing techniques (*e.g.*, squatting domains) and measured usages of the techniques in the wild. This motivates us to raise two fundamental research questions: (1) What is the strong drive for attackers who adopt HTTPS for phishing attacks? (see Section 4) (2) How well do CAs comply with required

practices? (see Section 5) (3) What are security implications when CAs do not follow best practices in the HTTPS phishing attack ecosystem? (see Section 6)

In this paper, we make the following three contributions.

• **Current landscape of HTTPS phishing attacks:** We seek to better understand the current landscape of HTTPS phishing attacks (see Section 4). Particularly, we take a closer look at the motivation of the HTTPS phishing attackers by measuring the effectiveness of HTTPS phishing attacks comparing to HTTP ones. We find that HTTPS phishing attacks account for 85.1% of the *successful* phishing campaigns. We believe that this high success rate and effectiveness would motivate the attackers to prefer HTTPS over HTTP.

• **In-depth analysis on practices of CAs:** As the issuance of valid certificates is an important factor in the HTTPS phishing attack, we investigate CAs—one of the crucial actors in the HTTPS ecosystem—to understand how their security practices can affect the HTTPS phishing ecosystem (see Section 5). In particular, we systematically conduct an experiment where we pretend a phishing attacker and interact with CAs (*e.g.*, applying for certificates and reporting them to the issuing CAs). We observe some security concerns which CAs do not comply with requirements [7] in issuance and revocation process cases. Specifically, they failed in checking high-risk (suspicious) domains and the same public keys that had previously been used for phishing attacks. Also, they rarely revoked abusive certificates in response to our reports to CAs.

• **Measurement of abusive certificates in the wild:** We aim to validate our observations from the empirical experiment in the wild (see Section 6). We use a longitudinal dataset of abusive certificates used for real phishing attacks in the wild. Our measurement results confirm the ecological validity of the experimental observations. Particularly, CAs rarely revoke abusive certificates and failed in checking, in the issuance process, high-risk domains and the same public keys that have been previously used for phishing attacks.

## 2 BACKGROUND

This section overviews the required practices and primary roles (*i.e.*, issuance and revocation) of CAs.

**Types of CAs.** Before 2016, there had been only a few commercial CAs (*e.g.*, GoDaddy, Comodo, DigiCert, etc.), and they required applicants (*e.g.*, server administrators) to pay for certificates. Due to the recent surge of need for the certificates, a new approach called Automated Certificate Management Environment (ACME) protocol was designed and deployed [27], in order to automate the issuance process of certificates. There are CAs utilizing the ACME protocol, *e.g.*, Let's Encrypt, to issue certificates at a very low cost, or even for free. Another way to obtain certificates is from *re-sellers*, who are authorized by CAs. They often provide the authorized CA's certificate at a lower price than the official CA.

### 2.1 Required Practices for CAs

**CA/Browser Baseline Requirements (CA/B BRs).** The Certification Authority Browser Forum is a voluntary business association of CAs, web browsers, and other PKI-related groups, aiming to publicize standard guidelines of the issuance and management of X.509 certificates. They published the current version of guidelines, CA/Browser Baseline Requirements (CA/B BRs) 1.7.2 in Sep. 2020 [7]. All CAs should comply with the Baseline Requirements in order to issue publicly trusted certificates.

**Certification Practice Statement (CPS).** Each CA publishes its own practice statements, called Certificate Practice Statement (CPS), that outlines the certification service practices: how to issue certificates (*e.g.*, verification) and when they revoke compromised/misused certificates. Each CA is required to perform such services in accordance with the CPS. Note that each CA explicitly mentions in their CPS that the CA/B BRs take precedence over their CPS if there is any conflict or inconsistency between their CPS and the CA/B BRs [10, 12, 13, 16, 21]. We, thus, present the roles and requirements of CAs based on the CA/B BRs in the rest of this paper.

### 2.2 Roles of CAs

CAs are responsible for (1) issuing certificates, (2) revoking certificates if compromised or misused, and (3) re-issuing certificates. We introduce the basic requirements of each certification service practice. Note that CA/B BRs defines that keywords (*e.g.*, "*SHALL*") should be interpreted in accordance with RFC 2219 [3]. Thus, in this paper, we interpret the keywords as the RFC specifies.

*2.2.1 Issuance.* The publicly trusted CAs can issue X.509 certificates that bind identities (*i.e.*, domain names) to cryptography keys that only applicants own. To issue the certificates, CAs have to verify the ownership of domain names (*i.e.*, Domain Validation).

**Cryptographic keys.** According to the CA/B Baseline Requirements, CAs shall reject a TLS certificate request (*i.e.*, certificate signing request) if a public key in the request is associated with known weak private keys [7]. CAs, also, shall check if the public key satisfies the minimum requirements; particularly in RSA, (1) the key length should be equal to 2048 bits or longer; and (2) for the public exponent ($e$), it should be the odd number greater and equal to 3 as well as between $2^{16}+1$ and $2^{256}-1$.

*2.2.2 Revocation.* CAs shall revoke their issued TLS certificates when they are informed that (1) the associated private keys have become compromised (*e.g.*, Heartbleed); (2) the issued certificates have been misused (*e.g.*, phishing attacks); or (3) certificates are fraudulently issued to adversaries (*e.g.*, DigiNotar) [11, 14, 20]. Moreover, when CAs are reported that their issued TLS certificates no longer comply with the CA/B Baseline Requirements (*e.g.*, a cryptographically weak key used for TLS certificates), they also shall revoke the certificates [7, 59].

*2.2.3 Re-issuance.* We define *re-issuance* as another issuance process where applicants re-apply for certificates that had been previously applied, but the certificates had been rejected or misused for malicious activities such as phishing attacks.

**History.** CAs shall maintain their own internal databases where they can keep track of all previously rejected certificate requests or revoked certificates for preventing suspicious certificate requests based on the Baseline Requirements [7][1]. Therefore, in their vetting

[1]The baseline requirements says that "In accordance with Section 5.5.2, the CA SHALL maintain an internal database of all previously revoked Certificates and previously rejected certificate requests due to suspected phishing or other fraudulent usage or

process, they shall check the public key in a certificate signing request (CSR) whether the public key had been misused for fraudulent or malicious purposes.

**High-risk domains.** CAs are recommended to have additional verification procedures for "*High Risk Certificate Requests* [7]."[2] High risk certificate requests include (1) domain names suspicious for phishing attacks or other fraudulent usages, (2) domain names of previously rejected certificate requests or revoked certificates, or (3) domain names listed on anti-phishing blacklists or the Google Safe Browsing list [7].

## 3 GOAL AND DATA COLLECTION

Phishing is a continuously critical security threat. Particularly, in recent years, the attackers have been adopting HTTPS, and the rapid surge of HTTPS phishing attacks is observed in the wild [23]. The attackers are required to obtain *valid* certificates from CAs in order to conduct HTTPS phishing attacks. This indicates that CAs are, at least indirectly, involved in the HTTPS phishing ecosystem because the attacks cannot be launched without valid certificates.

To mitigate the HTTPS phishing attacks, we first need to better understand how CAs are involved in the HTTPS phishing ecosystem. In particular, what insecure practices of CAs can lead to the increase of the attacks. However, unfortunately, the majority of prior work on phishing attacks has mainly focused on understanding the phishing techniques (*e.g.*, squatting domains) [34, 36, 46, 51, 56]. Therefore, little is known about how CAs are involved in HTTPS phishing attacks.

As the first step in this direction, our goal in this paper is to understand HTTPS phishing attacks by examining the security practices of CAs and how their practices could cause the surge of HTTPS phishing attacks, as outlined below.

- **The landscape of HTTPS phishing attacks.** We study HTTPS phishing threats' current landscape, including their changing trends and effectiveness compared with the HTTP phishing attacks, in order to identify important stakeholders and technical factors of the attacks (see Section 4).
- **In-depth analysis on practices of CAs.** We investigate Certificate Authority (CA)—one of the crucial but under-studied actors in the HTTPS ecosystem—to systematically check how they are involved in HTTPS phishing attacks. In particular, we conduct experiments where we play as a fake phishing attacker who interacts with the CAs in various aspects, including purchasing (*i.e.*, issuance and re-issuance), and reporting (*i.e.*, revocation) real certificates to better understand the consequences of CAs behaviors in the wild. Our experiments show that there are discrepancies between the expected security of CAs and reality (see Section 5).
- **Measurement of abusive certificates in the wild.** To validate our findings from the empirical experiment in the wild (Section 5), and investigate other security concerns, we conduct a longitudinal measurement study of abusive certificates used for phishing attacks in terms of issuance, revocation, and re-issuance. Our measurement results confirm the ecological validity of our findings. Moreover, we find new security concerns that cannot be observed from the empirical experiment (more details in Section 6).

concerns. The CA SHALL use this information to identify subsequent suspicious certificate requests."

[2]"The CA SHALL develop, maintain, and implement documented procedures that identify and require additional verification activity for High Risk Certificate Requests prior to the Certificate's approval, as reasonably necessary to ensure that such requests are properly verified under these Requirements."

Table 1: **Summary of our datasets.**

| URLs | FQDN | | Certificates | | |
|---|---|---|---|---|---|
| | Distinct | HTTPS | Total | Valid | Expired |
| 8,810,825 | 1,684,201 | 817,979 | 6,438,835 | 339,647 | 6,099,188 |

### 3.1 Dataset collection

**HTTPS phishing URLs.** To better understand the current landscape of HTTPS phishing attacks, we first collect URLs used for phishing attacks in the wild. We obtain unique 8,810,825 (8.81M) phishing URLs from `eCrimeX` [24], one of the largest and most trusted phishing URLs data repositories (*i.e.*, phishing blocklist) operated by the Anti-Phishing Working Group (APWG) [6]. The APWG, an industry association of anti-phishing entities, has collected the phishing URLs from its diverse member organizations. Our observation period covers 63 months, specifically from Jan. 1st, 2015 to Jun. 14th, 2020. These phishing URLs are used in Section 4.1 to understand the current trend of the HTTPS Phishing attacks.

**Fully Qualified Domain Names (FQDNs).** Before obtaining TLS certificates used for phishing attacks, we then need to extract FQDNs from the phishing URLs. This is because FQDNs are specified in the Common Name fields and Subject Alternative Name (SAN) fields of certificates. An FQDN syntax is defined in RFC 1035 [1]; particularly, an FQDN consists of three domains—a subdomain, a domain, and a top-level domain (*i.e.*, [subdomain].[domain].[top-level domain]).

Of the 8.81M phishing URLs, we find 1,684,201 (1.68M) valid, unique FQDNs in full compliance with RFC 1035 [1] (and later updates) as summarized in Table 1. It is very challenging to tell whether the FQDNs have been previously served with HTTPS since the FQDNs do not include any information regarding the network protocol. Therefore, we look for the TLS certificates of the FQDNs in the wild, and we then know whether the FQDNs are served with HTTPS. These valid FQDNs are used in Section 6 to understand which brands are being targeted for the squatting domains.

**TLS certificates.** The basic approach to look for TLS certificates in the wild is to access and download certificates directly from their websites as long as the website is alive and being served. However, we observe that many phishing websites become unavailable shortly after they are blocklisted [50], preventing us from downloading their certificates directly from their websites. Hence, we utilize *Certificate Transparency* (CT) where TLS certificates are logged for auditing purposes immediately after issued [40]. Since most major CAs participate in logging their TLS certificates in CT logs when issuing certificates, we are able to obtain all TLS certificates for those websites that are no longer alive at the time of data collection.

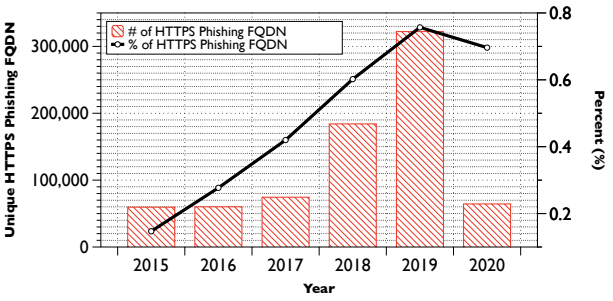


Figure 1: **HTTPS phishing domains by year.** Note that in 2020, the dataset was collected by June.

We obtain a total of 2,134,817,340 (2.1B) certificates from 13 CT logs[3] as of July 5th, 2020. We believe that these collected certificates can be a highly accurate representation in the wild because we utilize multiple CT logs of major CAs and of CT operators. Furthermore, VanderSloot et al. estimate multiple well-known CT logs can cover almost all certificates on the web [57]. We look for the TLS certificates used for the HTTPS phishing attacks. We check how many FQNDs that we extract from the 8.81M phishing URLs are found in 2.1B certificates. We observe that 817,979 FQDNs are found in the certificates, and their associated certificates are 6,438,835 (6.45M). On average, a phishing HTTPS FQDN has 7.9 TLS certificates. This dataset is used in Section 6.

## 4 HTTPS PHISHING ATTACK

In this section, we present the current landscape as well as the effectiveness of the HTTPS phishing attacks.

### 4.1 HTTPS Phishing Trend

Phishing is a fraudulent cyber operation that deceives users into believing a fake website is genuine, thus enticing the users to enter their private and sensitive information (*e.g.*, login credentials). In the APWG Trend Report Q2 2020 [23], it is reported that currently there are more HTTPS phishing websites than HTTP ones. When rendering an HTTPS website that has a valid certificate, most browsers display a *padlock* icon along with the website address. This padlock icon indicates the authenticity and integrity of a website. However, unfortunately, many users misunderstand the meaning of the padlock, misperceiving that any website with the padlock is trustworthy [32, 38, 43, 52]. Phishers are exploiting this misconception to initiate their attacks successfully [15, 18, 19, 44].

**HTTPS phishing URLs.** In the dataset of HTTPS phishing URLs from APWG, we observe 48.6% (817,979 out of 1.68M) of FQDNs have been used for HTTPS phishing attacks, which is slightly smaller than the number of HTTP phishing FQDNs. However, the 48.6% HTTPS FQDNs account for 66.7% of the total phishing attacks (5,879,710 URLs out of 8.81M), which indicates that HTTPS phishing FQDNs are used more repeatedly than the HTTP ones.

[3]Cloudflare (Cirrus, Nimbus 2017, Nimbus 2018, Nimbus 2019, and Nimbus 2020), Google (Argon 2017, Argon 2019, Argon 2020, Xenon 2020, and Rocketeer), Let's Encrypt (2019 and 2020) and Digicert Yeti 2020

Table 2: **Top-14 phishing target brands.**

| Target Brand | Alexa Rank | Total | HTTPS (%) |
|---|---|---|---|
| Facebook | 7 | 1,724,857 | 943,200 (54.7%) |
| Apple | 52 | 1,581,649 | 1,058,098 (66.9%) |
| Paypal | 101 | 500,864 | 272,683 (54.4%) |
| Yahoo | 10 | 468,467 | 189,990 (40.6%) |
| ChristianMingle | None | 573,764 | 155,123 (27.0%) |
| ATB | 28,760 | 330,202 | 162,346 (49.2%) |
| Bank of America | 336 | 297,882 | 189,990 (63.8%) |
| Microsoft | 22 | 282,912 | 233,668 (82.6%) |
| Wells Fargo | 169 | 176,049 | 81,686 (46.4%) |
| CIBC | 1533 | 165,152 | 91,291 (55.3%) |
| eBay | 40 | 151,065 | 21,372 (14.1%) |
| Others | | 6,494,972 | 1,958,984 (84.6%) |
| Undetermined | | 1,113,578 | 377,603 (33.9%) |
| Total | | 8,810,825 | 5,879,710 (66.7%) |

As shown in Figure 1, there is an increasing number of unique HTTPS phishing FQDNs every year—except in 2020 when our dataset was collected by June 2020. The percentage of HTTPS phishing FQDNs also rapidly rises every year as well. From 2018, the number of the attacks exponentially increased as the number of certificates issued from the Automated Certificate Management Environment (ACME) CAs—*e.g.*, Let's Encrypt—started to surge in the early of 2018 [25, 54]. We believe that the adversary can readily obtain free TLS certificates for their phishing attacks, leading to the rapid surge of HTTPS phishing attacks (more details in Section 6.2). In 2019, the volume of HTTPS phishing FQDNs (72.7%) is almost five times bigger than HTTP ones.

**Top phishing target brands.** We observe 4,063 targeted brands in the 8.81M phishing URLs of the APWG data repository. As shown in Table 2, the top-14 target brands account for 90.3% of phishing attacks—except for 1,113,578 phishing URLs (12.6% out of 8.81M) whose target brands cannot be determined. We find that most of the target brands are related to financial services. In addition, they are ranked in Alexa Top 1K domains except for `ChristianMingle`, an online dating website. Phishing URLs targeting the top-14 brands also use HTTPS (averagely 51.8%) more than HTTP. However, `eBay` has a significantly lower HTTPS phishing attack rate (14.1%) than the others. This is because most of the `eBay` phishing attacks (93.7% of 151,065) occurred in 2015 and 2016, where most phishing used HTTP.

### 4.2 Effectiveness of HTTPS Phishing

As shown in Figure 1, the number of HTTPS phishing attacks is rapidly increasing. However, the increased number of HTTPS phishing attacks does not necessarily mean the actual damage to the phishing victims by HTTPS phishing is more significant than HTTP phishing. To this end, we seek to measure the effectiveness of HTTPS phishing attacks. In this section, we provide an in-depth analysis of the *effectiveness* of HTTPS phishing websites by collaborating with a large online financial industry organization commonly targeted by phishing [56][4].

[4]As requested by the organization, we are unable to disclose the organization's name.

**Datasets & methodology.** To deepen our insight into the actual severity of the HTTPS phishing threat, in collaboration with the organization, a variety of data sources are used to identify suspicious domains, such as user reports of phishing e-mails, and ecosystem clearinghouses. Then, we collect traffic to live phishing websites from January 2020 to July 2020 using a recently proposed network monitoring framework which passively measures victim traffics [48].

Specifically, to gain an understanding of the aggregate volume of *successful* HTTPS phishing attacks, the framework first analyzes the internal web traffic logs of the organization to look for events of interest based on specific signals (*e.g.*, referrer headers with suspicious domains, and third-party resources embedded in phishing websites). Network events monitored by the framework have a high probability of being linked to a user targeted by a phishing attack. Moreover, certain events are a strong indicator of a successful attack: for example, a web visit to the organization's legitimate website, with a previously seen session cookie and a referrer header containing the URL of the last page of a known phishing website. We consider such events in our analysis.

**Findings.** We make several key observations.[5] *First*, 85.1% of successful phishing campaigns (monitored by the framework) used HTTPS while HTTP phishing attacks only accounted for 14.9%. *Second*, almost all of the phishing domains (43 out of the top 50 phishing domains) used HTTPS, and the top 50 phishing domains accounted for 44.6% of all victims. Much to our surprise, the results demonstrate that HTTPS phishing websites successfully fooled victims more than the traditional HTTP phishing attacks. These findings also suggest, at least in part, that the effectiveness would motivate the attackers to conduct HTTPS phishing attacks.

# 5 UNDERSTANDING CAS ECOSYSTEM

We have observed that the HTTPS phishing attacks are more effective, which could be a strong drive for the adversary. In this section, we systematically conduct an empirical study to investigate how the security practices of CAs impact the entire HTTPS ecosystem, specifically regarding HTTPS phishing attacks. To this end, we design experiments where we purchase TLS certificates from CAs and re-sellers, launch our own phishing websites with these issued certificates, report our abusive certificates to CAs or Google Safe Browsing (GSB), and measure their reactions to our phishing websites.

## 5.1 Phishing Websites Setup

**Phishing website construction.** For our phishing websites, we clone the login page of an online financial service, which is among the organizations most commonly targeted by phishers [56]. We then created a fake login page by copying the HTML code and images from the legitimate login page.

**Generating domain names.** It is critical to ensure that our experiments do not adversely affect legitimate users. We *randomly* generated domain names for our experiment (i.e., new domains that have never been registered before) so that benign users would be highly unlikely to access them. Also, these fresh, random domain names are not blocklisted, and thus, proper for this experiment.

**Hosting the phishing website.** We used `DigitalOcean` to host our phishing website. Then, we assigned static public IP addresses for each domain on a dedicated server. We reported our research plan to `DigitalOcean` to ensure that our experiments did not disrupt the infrastructure.

**Ethical concerns.** Users may visit our phishing websites by accident. In such a case, a user would inadvertently submit their personal information. To mitigate this issue, we carefully set up our controlled phishing websites with *random paths*, such that typical users would be extremely unlikely to directly access our phishing websites without knowing the exact full URLs. Moreover, we sterilize the submission form and remove the password field name, similar to an approach taken in prior empirical tests of the anti-phishing ecosystem [47]. Thus, no personal information is ever transferred or stored in the action of form submission. Additionally, it is possible that our experiments could interfere with CAs' ecosystem such as revocation of other abusive TLS certificates. To minimize the adverse side-effects for CAs, we limit the number of certificates we use for our experiments to 2 for each CA. This number is relatively small compared to the number of actually revoked certificates each day; on average, around 4,000 certificates are revoked a day in Sep. 2019 [8].

## 5.2 Experimental Design

Figure 2 shows the overview of our experiment. It consists of four steps as follows. ❶ For seven CAs and three re-sellers (totally ten), we apply for two TLS certificates to each CA, leading to a total of 20 certificate requests. ❷ We launch phishing websites with each issued certificate. To this end, we created 20 phishing websites. ❸ We report one of the abusive certificates to its issuing CA and another to Google Safe Browsing (GSB). If revoked, ❹ we re-apply for another certificate with the same public key used for the phishing site. Each step of our experiment investigates the *Issuance*, *Revocation*, and *Re-issuance* processes of a CA.

**Certificate types.** Note that we apply for only DV certificates: specifically, two for each CA with different configurations. This is because obtaining OV and EV certificates requires us to incorporate a shell company (or a legal business) and to submit the registration issued by the government to CAs, which can cause ethical issues.

**Selecting CAs for the experiment.** We choose the top-ten issuers (CAs) in our collected dataset (see Table 8) for our experiment because they account for 99.1% of the certificates of our collected datasets. We believe that the top-ten CAs dominate the Web PKI ecosystem. More information regarding the CAs is described in Table 6. Specifically, seven CAs include the `Comodo` group (`Comodo` and `cPanel`), the `DigiCert` group (`RapidSSL` and `GeoTrust`), `Let's Encrypt`, `Go Daddy`, and `GlobalSign`.[6] Of the top-ten CAs, we exclude CloudFlare and `DigiCert` because of the following reasons:

- `CloudFlare` is a web infrastructure company rather than a CA. The company is issued Subject Alternate Name (SAN) certificates [28] from CAs for websites hosted in their web service.

[5] We note that we cannot provide definite visibility of monitoring results because this requires exact knowledge of all phishing campaigns targeting the organization.

[6] Note that `COMODO` rebranded to `Sectigo`; thus only 7 CAs are selected.

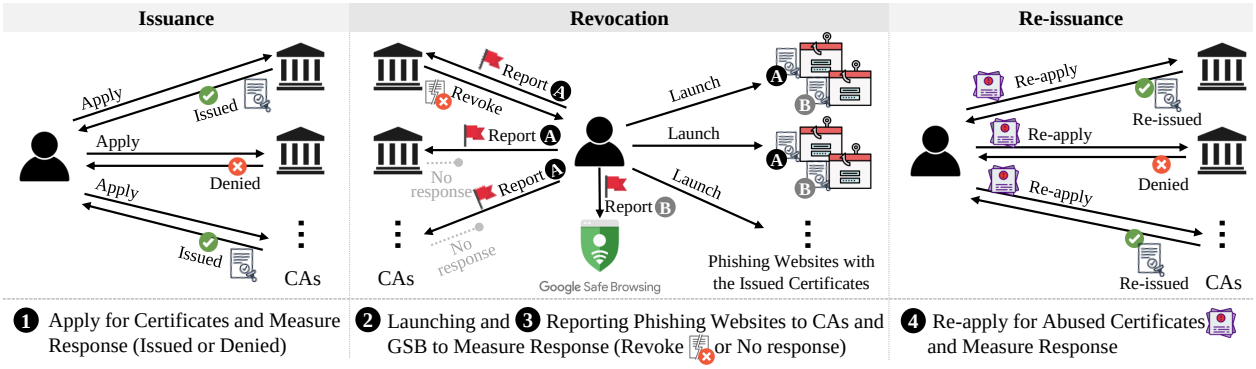


Figure 2: **Experimental design.**

Their SAN certificates are issued under the common name of `CloudFlare`, not an applicant's FQDNs (or domain names). Hence, we do not include `CloudFlare`'s certificates for our experiment because 1) the company is not a CA who issues a certificate and 2) when the applicants cannot control the certificates since `CloudFlare` owns and controls the certificates. Note that we investigate phishing attacks leveraging the SAN certificates in Section 6.2.

- `DigiCert` does not offer DV certificates unless a business partnership is established. This means that we may need to incorporate a company and to make a partnership agreement with the CA, which may lead to other ethic problems. Instead, we choose other CAs in the `DigiCert` group such as `RapidSSL` and `GeoTrust` that provide DV certificates.

**Certificate re-sellers.** In addition to the official CAs, we also purchase certificates from re-sellers because they are another channel for certificate applicants (including benign and malicious users). Thus, we also want to understand how they affect the ecosystem of the Web PKI. Three re-sellers include `comodosslstore` for `Comodo`, `thesslstore` for `RapidSSL`, and `cheapsslsecurity` for `GeoTrust`.

#### 5.2.1 *Experimental Procedure.*

**1 Issuance.** We aim to investigate whether we can obtain certificates with insecure configurations for public keys, which should be prohibited by CAs if they comply with the baseline requirements [7]. Specifically, in the certificate issuance process, CAs are responsible for checking if requested public keys meet the requirements. For example, as discussed Section 2.2, the public exponent ($e$) in the RSA algorithm should be carefully chosen because the algorithm is vulnerable (*e.g.*, plain data can be exposed) when the exponent is '3' [29, 58]. We consider only RSA in our experiments because RSA public keys dominate (more than 99%) in trusted leaf certificates in the Web PKI [30]. In our experiment, we apply to the ten CAs (including seven CAs and three re-sellers) for TLS certificates with two types of RSA weak keys: (1) 1024 bits or (2) $e = 3$. If we are successfully issued certificates with weak RSA keys, CAs may fail in complying with the baseline requirements.

**2–3 Revocation.** We leverage the certificates that we obtained in the previous issuance experiment. Specifically, we obtain two certificates from each CA (the seven CAs and three re-sellers), resulting in 20 certificates. We then launch 20 phishing websites using our phishing kit (see Section 5.1) with the issued certificates. Consequently, we have two phishing websites associated with each CA or re-seller. Then, we report (1) one of the two certificates directly to the issuing CA and (2) another one to only GSB.

(1) From directly reporting to its issuing CAs, we aim to measure how promptly the CAs take actions, such as the total time spent on the investigation of reported phishing URLs and on the revocation of certificates used for the phishing domains. We also measure how CAs respond to the reporters or the owners of the certificates. Moreover, we measure how CAs investigate on our reported certificates used for phishing attacks by analyzing the access logs of our Web servers.

(2) By reporting only to GSB, we aim to understand whether or not CAs proactively investigate phishing websites by leveraging other available phishing datasets such as GSB and revoke them. Moreover, if they take actions proactively (*i.e.*, revoking certificates), we further measure how promptly those actions had been taken.

**4 Re-issuance.** We aim to better understand whether CAs re-issue certificates to suspicious applicants who are likely to conduct phishing attacks.

- ***Same public keys:*** For the certificates revoked by CAs in the previous revocation experiment (3), we conduct an additional experiment to investigate whether CAs can remember the revoked certificates or public keys used in them to prevent re-issuance attempts from malicious actors. Hence, we *re-apply* for certificates with the same public keys used in the revoked certificates. Adversaries are less motivated to behave in this way, but note that this experiment is necessary to better the security practices of CAs in their issuance process.
  According to the Baseline Requirements [7], it explicitly mentions that *CAs shall maintain internal databases and keep track of rejected certificate requests or revoked certificates for fraudulent usages such as phishing websites* (see Section 2.2). Note that applying a new certificate with the same public key used in a phishing attack suggests that the application comes likely from the same attacker. A desired outcome is to have additional scrutiny or reject certificate requests with the same public keys used for phishing attacks and in revoked ones.

- ***High-risk domain for a real phishing attack:*** Moreover, we try to obtain certificates of suspicious domains from CAs. Specifically, we use a domain name that was previously used for a real phishing attack in our dataset (see Section 3.1). It is a typical squatting domain (*pyp-al.com*) and the domain was already blocklisted in Google Safe Browsing in Oct. 2019. We can take over the domain since it becomes available for registration. We then apply for a certificate to each CA with the high-risk domain name. Note that as discussed in Section 2.2, CAs are expected to perform an additional strict verification process when requested domains look suspicious (*e.g.*, if the domains are known to be used in phishing attacks). Hence, during this experiment, we expect CAs to conduct some additional vetting processes on our issuance requests.

## 5.3 Experimental Results

We present experimental results of the three processes: (1) issuance, (2) revocation, and (3) re-issuance of CAs, as described in Table 3.

*5.3.1 Issuance.*

**Weak public key sizes.** All CAs and re-sellers properly checked the key length of RSA public keys (*i.e.*, whether the keys are greater than or equal to 2048) in their online form where applicants put their CSRs (certificate signing requests), and rejected CSRs with the key length less than 2048 bits (*e.g.*, 1024 bits) which is considered as a weak key, meaning that they comply with the requirements.

**Vulnerable public exponents.** We find the `DigiCert` group (`RapidSSL` and `GeoTrust`) and `GlobalSign` *improperly* check the security requirements of the public exponent $e$ in the RSA algorithm in CSRs. Specifically, while $e$ should be an odd number larger or equal to 3 and between $2^{16}+1$ and $2^{256}-1$, they have issued certificates with $e = 3$, which can lead to a security vulnerability [29, 58].

*5.3.2 Revocation.* We observe that the results of the revocation experiment depend on CAs or parent CAs of the re-sellers because the issued certificates are managed by the issuing CAs or the re-sellers' parent CAs. Hence, in the following paragraphs, we present the result by group or each CA. Recall that in this experiment, for each group or CA, we report the URLs of our phishing websites (which use the certificates issued from each group or CA) to the responsible CAs (*e.g.*, the CA that issued the certificate) and Google Safe Browsing (GSB) as shown in Figure 2 (A) and (B). In this subsection, note that '$***$' is appended to the re-sellers' names for easy recognition.

**Comodo group.** The `Comodo` group includes `Comodo` (now renamed as `Sectigo`), `comodosslstore`$^{***}$, and `cPanel`. `cPanel` is an automated CA that has a partnership with `Comodo`, providing certificates under the `Comodo`'s brand. The certificates issued by the `Comodo` group share the same revocation policy which says they will revoke the certificates used for illegal or fraudulent activity (see Table 9). We reported the phishing websites to "ssl_abuse@sectigo.com." We received responses from them between *16 hours to 3.5 days*, saying that the reported URLs and certificates are under investigation. However, as of Sep. 1st, 2020 (more than 11 months later), the certificates are *not revoked*.

**DigiCert group.** `DigiCert` group includes `RapidSSL`, `GeoTrust`, `thesslstore`$^{***}$, and `cheapsslsecurity`$^{***}$. We reported the phishing websites to the two CAs via their official web pages[7] and GSB. We received responses from the two CAs within *one hour* regarding the reports, mentioning that they have begun their investigations. From our web server's access logs, we find three visitors' IPs: our own IP, `DigiCert`'s IP, and VirusTotal [22] crawling bot's IP in this particular order. The IP logs indicate that the `DigiCert` investigation team might utilize VirusTotal to query for our reported URLs for investigation. Fortunately, they have revoked our abusive certificates within approximately 24 hours. However, regarding other certificates used in the phishing websites reported to GSB, they are not revoked as of Sep. 1st, 2020 (more than 11 months). In short, the `DigiCert` group promptly investigates the reported certificates and revokes them within 24 hours, complying with their CPS while they may not monitor third party databases (*e.g.*, GSB) to identify abused certificates—note that such proactive investigation is not a requirement for CAs in terms of revocation.

**Let's Encrypt.** We reported our phishing website to cert-prob-reports@letsencrypt.org and the other phishing website to GSB. They will revoke certificates used in cyber crimes according to their CPS [16] (see Table 9). We received automated CA responses to our report within *one hour*. However, we find that their response does not comply with their CPS. Their email response states that "*our (`Let's Encrypt`) policy does not allow us to revoke certificates for sites suspected of engaging in phishing, distribution of malware, or other forms of fraud*," which directly may conflict with their CPS which mentions "entitled to revoke" as described in Table 9. They also ask us to report the phishing website to GSB by ourselves, mentioning that "*GSB is able to more effectively protect users*." In short, `Let's Encrypt` did not actively involve the management of certificates such as revocation. Note that `Let's Encrypt` is an automated CA, providing TLS certificates without any human intervention for free. From the phishing attackers' perspective, the automated CA might be a preferable option for them, considering it is easy and cheap to obtain certificates and those certificates might be rarely revoked.

**Go Daddy.** We also reported our phishing website to both the report form[8] and GSB. Their CPS explicitly mentions that they will provide a preliminary report to the filed report [13] (see Table 9). We *did not receive any responses or investigation results* from the CA. Due to no response, we are unable to know whether or not the CA has started investigations on the abusive certificates. When we check the revocation status of the reported certificates, the certificates are not revoked as of Sep. 1st, 2020 (more than 11 months).

**GlobalSign.** We reported our phishing websites to their abuse report form[9] as well as to GSB. We received their automated response within *one hour*, saying they would start their investigation on the reported URL and certificate. In 44 hours after the reporting, we received their investigation result that claims "*the URL and certificate have not committed any violation of GlobalSign Subscriber Agreement and Certification Practices Statement (CPS)*". However, they did not provide any further information regarding their conclusion. *Nine days later* after we reported our certificates, we have received

[7] https://www.rapidssl.com/contact/ssl-certificate-complaint.html and https://www.geotrust.com/about/contact/ssl-certificate-complaint.html
[8] https://supportcenter.godaddy.com/AbuseReport
[9] https://www.globalsign.com/en/report-abuse

Table 3: **Experimental results.** No CAs properly check high-risk certificate requests. The `DigiCert` group and `GlobalSign` have issued certificates with RSA weak keys. Only `GlobalSign` and the `DigiCert` group revoke our certificates reported only to its issuing CA. All CAs except for `GlobalSign` never revoke abusive certificates reported only to a third-party (*e.g.*, GSB).

| Certificate Authority | | Issuance | | Revocation | | | Re-Issuance | |
|---|---|---|---|---|---|---|---|---|
| Group | CA | Key size* | Pub. exp.** | Resp. delay | Revc. delay | Proactive | Revk. Key | High Risk |
| Comodo | Comodo | ✓ | ✓ | ≈16h | ✗ | ✗ | N/A | ✗ |
| | comodosslstore*** | ✓ | ✓ | ≈3.5d | ✗ | ✗ | N/A | ✗ |
| | cPanel | ✓ | ✓ | ≈3.0d | ✗ | ✗ | N/A | ✗ |
| DigiCert | RapidSSL | ✓ | ✗ | ≈1h | ≈1d | ✗ | ✗ | ✗ |
| | thesslstore*** | ✓ | ✗ | ≈4h | ≈1d | ✗ | ✗ | ✗ |
| | GeoTrust | ✓ | ✗ | ≈5h | ≈1d | ✗ | ✗ | ✗ |
| | cheapsslsecurity*** | ✓ | ✗ | ≈2h | ≈1d | ✗ | N/A | ✗ |
| Let's Encrypt | | ✓ | ✓ | ≈1h | ✗ | ✗ | ✗ | ✗ |
| Go Daddy | | ✓ | ✓ | ✗ | ✗ | ✗ | N/A | ✗ |
| GlobalSign | | ✓ | ✗ | ≈1h | ≈17d | ≈17d | ✗ | ✗ |

*RSA key size ≥ 2048-bit. **Public exponent $e$: odd & ≥ 3 & (between $2^{16}+1$ and $2^{256}-1$). ***: re-sellers.
✓: Vulnerable key size and pubic exponent are checked during the Issuance experiment.
✗: Vulnerable key size and pubic exponent are not checked during the Issuance experiment, a certificate was not revoked during the Revocation experiment, an abused certificate was not checked, re-issuing an abusive certificate during the Re-issuance experiment.

two emails from `GlobalSign`, mentioning that they received complaints from (unrevealed) third-parties that our two certificates have been misused in phishing websites. The email also mentions that *if we do not provide proper vindication within 24 hours, they will revoke the certificate*. We did not provide any feedback to the CA intentionally, expecting them to revoke the certificates. After another *week* passed, the two certificates were eventually *revoked on Oct. 14th, 2019*; they took *17 days* to revoke the two certificates after we report them. In short, `GlobalSign` monitored their issued certificates and third-party databases, although performing a revocation of an abusive certificate took a week that is *longer* than the deadline defined by the Baseline Requirements [7].

#### 5.3.3 *Re-issuance.*

**Public key used for phishing attack.** CAs shall maintain their own internal databases that keep track of revoked certificates or public keys used in the revoked certificates to prevent re-issuance of already abused certificates according to the Baseline Requirements [7] (see Section 2.2).

To measure whether the requirement is properly practiced, we *re-apply* for certificates to each CA with the same public key of the certificates used for our phishing websites only when the CAs *had revoked* the certificates, meaning that the CAs are aware of the public key used for phishing attacks. Note that in our experiment design, we cannot re-apply to all CAs because our all abusive certificates are not revoked, and we are interested in only revoked certificates for this experiment. Moreover, since we can revoke a certificate issued from `Let's Encrypt` by ourselves with the private key, we intentionally revoke the certificate and re-apply with the same public key of the revoked certificate.

Our requests with the already-abused public key (*i.e.*, the key used for previously revoked certificates) are passed CAs' vetting processes, successfully being issued certificates from CAs including `Let's Encrypt`, `GlobalSign`, `RapidSSL` (both official and thesslstore), and `GeoTrust` (official). Note that we face a technical problem during the experiment with `cheapsslsecurity`; the problem was caused because the verification of the domain ownership over the DNS records using certificate authority authorization (CAA) failed for unknown reasons[10]. We specify it as N/A in Table 3.

The result suggests that the `DigiCert` group, `GlobalSign`, and `Let's Encrypt` might not keep tracking the record of rejected requests or revoked certificates to mitigate the re-issuance problem of the abused certificates.

**High-risk domain used for real phishing attack.** According to the CA/B Baseline Requirements [7], CAs are *responsible* for checking if a certificate request may include names at higher risk for phishing or other fraudulent usages[11]. In this experiment, we use *pyp-al.com* used for a real phishing attack and blocklisted on GSB. We expect we would be asked for additional scrutiny if we request a certificate with the suspicious domain used in an already-reported phishing attack.

Unfortunately, as described in Table 3, all CAs and re-sellers have issued certificates for the high-risk domain *without further verification process* after they just checked the ownership of the domain; the CT logs are shown in Table 7. During the issuance process, we (applicant) were never asked to provide more information for additional investigation, suggesting that the CAs may not check whether or not a requested domain is suspicious.

## 6 ABUSIVE CERTIFICATES IN THE WILD

We have observed that the current practices of CAs from our experiment raise several security concerns. In this section, for the ecological validity, we want to know if our findings in the three

[10] We contacted to `cheapsslsecurity` to resolve the issue, but they were not able to resolve as well.
[11] It mentions that CAs are responsible for checking if a certificate request "include names at higher risk for phishing or other fraudulent usage, ... , names listed on the Miler Smiles phishing list or the Google Safe Browsing list"

roles—*i.e.*, *issuance* (Section 5.3.1), *revocation* (Section 5.3.2), and *re-issuance* (Section 5.3.3)—are valid in the wild using the longitudinal datasets of abusive certificates. Moreover, we examine more security concerns that we cannot measure from the empirical experiment.

## 6.1 Refining Datasets

To better understand how adversaries interact with CAs (*e.g.*, issuance, revocation, re-issuance), we need to examine abusive certificates that are directly issued to the adversaries and managed by them. To this end, we exclude the following three cases from our datasets because we believe adversaries cannot interact with CAs if one of the three cases occurs.

(1) `CloudFlare`: Attackers use the web hosting or CDN services from `Cloudflare` for their phishing websites. In this case, the attackers do not interact with CAs at all—*e.g.*, attackers do not submit certificate signing requests (CSRs) to CAs; rather, `CloudFlare` applies for and manages certificates.

(2) Free services: Attackers frequently employ free services including free web hosting services (*e.g.*, `000webhost.com`) or free blog services (*e.g.*, `blogspot.com`) to launch phishing attacks. In this case, the phishing web pages can be served with HTTPS using wildcard certificates of the service providers, and the attackers are not involved in the issuance.

(3) Compromised web servers: While it is challenging, attackers can launch phishing attacks by exploiting vulnerabilities of HTTPS web servers or web applications such as `WordPress`. Hence, albeit certificates are not issued directly to the attackers, the phishing attacks are served with HTTPS.

We believe that the free services are likely ranked in the Alexa top-1M domains due to its popularity. We use Alexa Top 1M domains to exclude the phishing FQDNs served on free services or on compromised web servers. Moreover, to be more conservative, we also include two additional top 1M domains from Cisco [5] and Majestic [17]. In total, we find 16,425 domains are observed in the three 1M domains and exclude them from our dataset, which remains 574,036 distinct FQDNs and 5,739,338 certificates. For the last step where we need to filter out certificates issued to `CloudFlare`, we remove certificates if their subject `commonName` contains `cloudflare.com` or `cloudflaressl.com`. After this, eventually we have 527,793 distinct FQDNs with 3,437,605 certificates.

## 6.2 Issuance

**RSA public key.** We examine how CAs follow the Baseline Requirement and properly issue certificates in terms of RSA public key size and public exponent.

(1) *Key size*: We observe no certificates that have RSA public keys smaller than 2048 bits after 2014. This is in line with our experiment (see Section 5.3) where all the CAs properly checked the RSA public key sizes and they successfully rejected certificate requests containing smaller than 2048 bits. Moreover, for ECDSA public keys, all issued certificates in the wild follow the Baseline Requirement; they all are secure.

(2) *Public exponent*: 99.9% of certificates in the wild have recommended public exponents while we find a very small number of certificates whose RSA exponents do not meet the guideline. The high number of proper public exponents points out that the CAs also correctly meet the requirements, which is the same as the case of the key size.

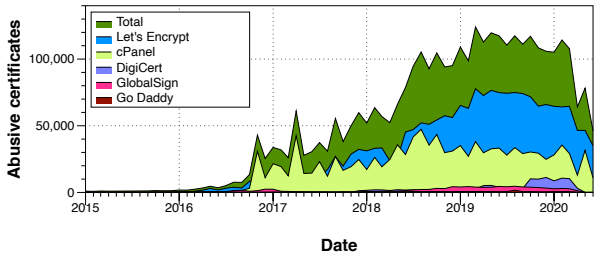


Figure 3: **Abusive certificates over year broken down by the top-five issuing CAs.**

**Top issuers (CAs).** We believe the advent of automated CAs (`Let's Encrypt` and `cPanel`) brought the rapid surge of abusive certificates for phishing attacks. Figure 3 shows our measurement results on the number of distinct abusive certificates issued by each CA over time. We confirm that attackers use the automated CAs to obtain certificates much more than commercial CAs. Specifically, 91.6% have been issued from the free, automated CAs while only 8.4% were from commercial CAs as shown in Table 4. The automated CAs can be one of the main contributors to the surge of the HTTPS phishing attacks due to low cost. We believe that the automated CAs may need to have more responsibility for managing certificates as well as issuance.

**SAN certificates.** A SAN (Subject Alternate Name) certificate allows having multiple FQDNs (domains) that can be secured by a single certificate. We examine how many abusive certificates have more than two FQDNs used for phishing attacks in the wild. In this scenario, the attackers can apply for a single SAN certificate and include multiple FQDNs that will be used for their multiple phishing campaigns.

We look for SAN certificates that contain more than two phishing FQDNs. In our refined dataset, we discover 76,363 SAN certificates (6.69% of 1,142,078 SAN certificates) and these certificates have 234,915 HTTPS phishing domains reported to the APWG. This indicates that averagely 3.1 phishing domains are included in a single SAN certificate.

Moreover, we also consider another scenario where an adversary applies for a SAN certificate with a benign FQDN for its common name, but they specify malicious FQDNs for a phishing attack in the SAN field. We observe that there are 280,011 SAN certificates (24.51% of of 1,142,078 SAN certificates) — associated with 342,258 HTTPS phishing domains—of which common names have not been reported but its SAN domains in the SAN field are reported and flagged as malicious domains. This attack scenario is more observed in the wild than the former scenario, which indicates that the attackers more frequently use benign common names when they request SAN certificates for phishing attacks.

## 6.3 Revocation

We observe in our experiment (see Section 5.3) that CAs rarely revoked certificates used for phishing websites. We examine how

Table 4: **Top-10 Issuers (CAs) excluding CloudFlare.** We checked the revocation status on Sep. 22, 2020. We cannot check the status of 23 certificates due to OCSP errors [35].

| Cat. | Issuer (CA) | Total | Status | | |
|---|---|---|---|---|---|
| | | | Good | Expired | Revoked |
| A | Let's Encrypt | 1,951,718 | 130,708 | 1,820,993 | 17 (0.01%) |
| A | cPanel | 1,197,283 | 57,710 | 1,139,566 | 7 (0.01%) |
| C | GlobalSign | 102,248 | 17,453 | 84,645 | 150 (0.85%) |
| C | Go Daddy | 34,175 | 7,458 | 24,937 | 1,780 (19.27%) |
| C | COMODO | 31,523 | 1,345 | 30,133 | 45 (3.24%) |
| C | Sectigo | 30,233 | 11,355 | 18,617 | 261 (2.25%) |
| C | DigiCert | 19,802 | 8,715 | 11,003 | 84 (0.95%) |
| C | RapidSSL | 9,772 | 1,300 | 8,465 | 7 (0.54%) |
| C | GeoTrust | 5,857 | 855 | 4,995 | 7 (0.81%) |
| C | Entrust | 5,350 | 4329 | 995 | 26 (0.60%) |
| | Etc. | 49,644 | 8,146 | 40,987 | 511 (5.90%) |
| | **Total** | 3,437,605 | 249,374 | 3,185,336 | 2,895 (1.15%) |

A: free, automated CA, C: commercial CA.

CAs properly revoke abusive certificates issued for real phishing attacks in the wild. To check the revocation status of abusive certificates in our dataset, we use Online Certificate Status Protocol (OCSP) [53]. As shown in Table 4, on average, only 1.15% of abusive certificates have been revoked by the CAs. This indicates that CAs rarely revoked certificates similar to our experiment. Particularly, automated CAs averagely have lower revocation rates (0.01%) comparing to commercial CAs (6.88%).

CAs can start investigations on misused certificates proactively or/and reactively, and may revoke the certificates if necessary. The extremely low revocation rate basically suggests that CAs themselves do not proactively monitor the blocklists (*e.g.*, eCrimeX or GSB) to check if their issued certificates are maliciously used for phishing attacks, or/and they do not actively revoke reported abusive certificates. Therefore, even though phishing websites have a very short life cycle [48], users will remain exposed to HTTPS phishing attacks because adversaries can repeat the attacks using the same domains and certificates as far as the certificates are valid.

**Importance of revocation.** Prior work [32, 38, 43, 52, 55] has shown that the HTTPS indicator (*e.g.*, padlock icon) on the web browser's address bar may mislead users to believe even phishing websites are legitimate and safe. We believe that phishing attacks with no HTTPS indicator (padlock) might not be as effective as the phishing attacks with the HTTPS indicator. Unfortunately, the abusive certificates' low revocation rate raises a concern that known abusive certificates still mislead users helping attackers. Moreover, we observe that 92.7% of the certificates have naturally been expired. However, those certificates were expired after *at least 90 days*, while most phishing websites last only a couple of days (specifically, 61.69 hours) [45]. This suggests that the certification expiration policy may not be as effective as the revocation.

## 6.4 Re-Issuance

In Section 5.3.3, we found that the CAs failed in checking (1) the high-risk domains and (2) the public keys used for phishing attacks when re-issuing certificates. We further analyze the re-issuing process CAs via our refined datasets.

*6.4.1 High-Risk: Squatting Domains.* Recall that *domain squatting* is one of the most common techniques for phishing attackers to trick victims. End-users, who may have just glanced at the browser's address bar, can believe the squatting domains to be legitimate. We specifically find squatting domains *reported* as phishing domains and *re-issued* by CAs.

**Squatting techniques.** We briefly explain three popular squatting techniques and find squatting domains in our datasets based on them.

- *Combosquatting:* A combosquatting domain is composed by appending additional keywords to the beginning or the end of the target domain. For example, to create a combosquatting domain of `apple.com`, an adversary can append "`www2-login-appleid-`" in front of the domain's effective second level domain name (e2LD) which is `apple`, resulting in `www2-login-appleid-apple.com`, which is a real example in our dataset.
- *Typosquatting:* Typosquatting attacks take advantage of the typos from the end-users by registering phishing domains that are very similar to the target domain. For example, from `apple.com`, a character can be omitted (*e.g.*, `aple.com`), permuted (*e.g.*, `aplpe.com`), substituted (*e.g.*, `appke.com`), or duplicated (*e.g.*, `appplle.com`).
- *Homograph:* An adversary can exploit the fact that some characters look alike; in English, '`l`' (L) and '`1`' (the number) look similar. For example, "`app1e.com`" can be registered where "`l` (L)" is replaced with "`1`."

**Target brand selection.** Phishing attackers imitate the popular brand names through squatting techniques to lure more victims. We utilize the Alexa Top 1K domains [4] for generating squatting domains, similar to prior works [26, 37, 51]. In the Top 1K domains, some brands may have multiple top-level domains (TLDs)—*e.g.*, `amazon.com`, `amazon.in`, and `amazon.co.jp`. We, thus, merge the brands that have the same domain names, but different TLDs. Moreover, to extend the coverage, we also use the 702 unique brands selected by Tian et al. [56]. These brands are chosen from Alexa's 17 categories—*e.g.*, "games," "health," "business," which helps us cover a wider range of top brands on the Internet. By merging the two datasets, in total, we have unique 1,345 top brands (*i.e.*, domain names). We use the domain names to find out squatting domains used for the HTTPS phishing attacks.

**Identification.** We first employ DNSTwist [2] to generate squatting domains through typosquatting, homograph, and other techniques. In addition, we follow the same methodology that Kintis et al. [36] proposed to generate squatting domains by using combosquatting technique. First, we extract all e2LDs from our target top brands (*e.g.*, "`paypal`" in "`paypal.com`"). Then, we check if the e2LDs (effective second-level domains) are a substring of our phishing dataset. However, the top brand names can be substrings of English words (*e.g.*, `vice.com`'s e2LD is `vice`, which is a substring of `service`). Therefore, to be more conservative, we first filter out all target brand domains whose e2LDs can become substrings of English words by using the English dictionary.[12] Furthermore, we manually inspect and remove target brand names to check if they can be a substring of English words.

[12] https://github.com/dwyl/english-words

Table 5: **Top-10 Squatting domains.**

| Target Domain | Rank* | Total | Squatting Type | | | |
|---|---|---|---|---|---|---|
| | | | Combo. | Homo. | Typo. | Others |
| Apple | 52 | 27,611 | 27,248 | 132 | 145 | 86 |
| Paypal | 101 | 8,268 | 7,968 | 89 | 64 | 147 |
| LinkedIn | 58 | 6,000 | 5,926 | 6 | 19 | 49 |
| Chase | 117 | 2,040 | 2,016 | 1 | 4 | 19 |
| Bank of America | 336 | 1,456 | 1,442 | 4 | 3 | 6 |
| Facebook | 7 | 1,358 | 1,304 | 11 | 7 | 36 |
| Airbnb | 802 | 977 | 951 | 14 | 5 | 7 |
| Amazon | 12 | 807 | 770 | 4 | 12 | 21 |
| WhatsApp | 70 | 799 | 794 | 4 | 2 | 3 |
| Netflix | 21 | 498 | 173 | 13 | 6 | 6 |
| Etc. | | 27,574 | 18,212 | 542 | 2,396 | 3,438 |
| **Total** | | 75,440 | 67,105 | 817 | 2,663 | 3,818 |

*: Alexa Top Rank

**Results.** Table 5 shows the measurement results. With the 1,345 top brands, we observe that 75,440 HTTPS FQDNs (14.3% out of 527K HTTPS FQDNs) are squatting domains in our dataset. We observe that phishing attackers use squatting techniques to imitate the most-targeted brand names in phishing (see Section 4.1). Moreover, we find that the attackers use the *Combosquatting* technique more than the others. Specifically, 88.9% of squatting FQDNs (67,104 out of 75K) were generated through the combosquatting technique. 351,234 certificates are issued with the 75K FQDNs; averagely, each HTTPS FQDN has 4.7 certificates. This observation is in line with our empirical experiment, specifically re-issuance process (see Section 5.3.3) where we found that the certificates of high-risk (suspicious) squatting domains were successfully issued.

*6.4.2 High-Risk: Re-used Phishing Domains.* We investigate if CAs allow the re-issuance of the TLS certificates whose domains had been reported. We believe the certificates to be abusive because they contain the reported malicious domains. We first categorize the certificates into two groups using the validity periods (see Table 6) in order to better understand them: (1) re-issued ones that the automated CAs (*e.g.*, `Let's Encrypt` and `cPanel`) issued (90 days); and (2) the others (365 days).

We observe that CAs allowed re-issuance of 248,514 certificates—among them, 110,710 issued by the automated CAs and 137,804 issued by other CAs. Particularly, phishing certificates that `Let's Encrypt` issued were re-issued 803,278 times in total, and each certificate was re-issued about 7.3 times within 68.4 days on average. The other phishing certificates were re-issued 977,308 times in total. These certificates were re-issued 7.1 times within 147.8 days on average. Figure 4 shows the cumulative distribution function (CDF) of re-issued phishing certificates for each group. Understandably, adversaries requested re-issuance of roughly 80% of re-issued certificates within 90 days (before the certificates expire). On the other hand, about 80% of certificates were re-issued by the other CAs within 365 days. These measurement results also indicate that the abusive certificates are rarely revoked, which helps the attackers to be able to re-apply for new ones with the same phishing domains when approaching their certificates' expiration date. We believe that the poor revocation performance can be a major reason.

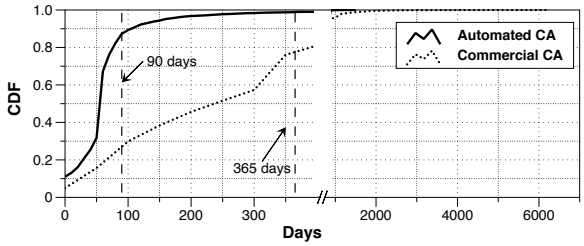


Figure 4: **CDFs of re-issued certificates by CAs.**

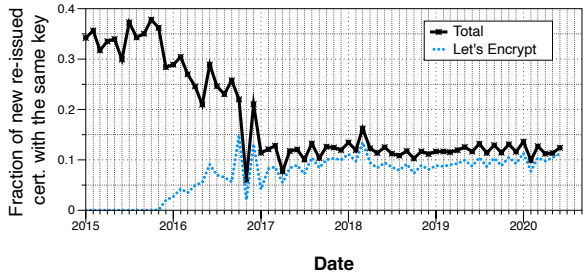


Figure 5: **Fraction of newly re-issued certificates with the same public keys.**

*6.4.3 Re-issued with the Same Public Keys.* It is mathematically very challenging to regenerate the same private key from a public key if the key is long enough [42]. If there are two certificates that share the same public key, the two certificates belong to a single person (or a group) even though the certificates have different common names. In other words, the re-issuing with the same public/private key pair can be a fingerprint of the phishing attackers. Based on this fact, we can retrace the course of the attackers' misbehavior.

We observe that before 2017, the phishing attackers re-used their public/private key pairs when applying for new certificates as shown in Figure 5. This finding is in line with the prior work [60] that found that almost half of the benign certificate (of the Alexa top-1M domains) is re-issued with the same public keys before 2014. This indicates that the phishing attackers behave like the benign web administrators in terms of re-using the same public/private key pair. Moreover, after 2017, the re-usage rate goes down to approximately 10%. This is at least partially because of the attackers who use `Let's Encrypt`.

We find 120,007 unique public keys that have been used for more than two different certificates. The public keys are associated with the 419,842 certificates (12% out of 3.4M certificates); each public key is averagely used for 3.5 certificates. Particularly, a single public key is used for 182 phishing domains; it was first used on April 26, 2018 and have been used until Jan. 4, 2020. On average, the same public keys are used for 150.7 days. The longest usage of the same public key is 3592 days (9.8 years). Specifically, the attacker used the public key occasionally; he/she first used it in April 2010 and still used in Feb. 2020.

## 7 RELATED WORK

**Phishing Attacks.** Measurement studies about phishing attacks have been well studied [34, 36, 46, 51, 56]. Particularly, their research work mainly focused on understanding squatting phishing domains and the entire phishing ecosystem. They barely discussed abusive certificates used for phishing attacks except for Roberts et al. [51] that conducted a measurement study of domain squatting attacks in TLS abusive phishing certificates. However, none of the research works understood the CAs ecosystem by systematically interacting the CAs what security concerns they currently have. In our work, we systematically measured the CAs' security practices and found the security concerns that may lead to the surge of the HTTPS phishing attacks.

**Primary Roles of CAs.** CAs are responsible for the primary roles: (1) issuance (including re-issuance) and (2) revocation. (1) For issuance, Kumar et al. developed a linting framework, called ZLint to quantify the misissuance of TLS certificates [39]. They found that large CAs correctly follow the requirements to issue TLS certificates; but the small CAs sometimes issued incorrect TLS certificates. This research work mainly focused on grammatical misissuance (*i.e.*, certificates with errors) such as "ExtKeyUsage not critical." In contrast, we measure if CAs follow the Baseline Requirements and their CPS in terms of RSA weak keys and high-risk (suspicious) certificate requests by systematically conducting experiments where we interact with the real CAs (*e.g.*, purchasing certificates directly from CAs). (2) Revocation in the Web PKI has been well studied. Particularly, Durumeric et al. and Zhang et al. measured the revocation rate after the Heartbleed OpenSSL bug was disclosed [31, 60]. Liu et al. also measured the certificate revocation in the wild [41]. These research works merely focused on measuring the bad practices of web server administrators when they have to revoke their certificates after a critical bug had been disclosed, or how many general TLS certificates were revoked. Contrarily, we mainly focus on the abusive phishing certificates by interacting with real CAs (*e.g.*, how promptly CAs revoke abusive certificates after the CAs are notified).

## 8 CONCLUSION

As a rapid surge in the number of HTTPS phishing attacks, the roles of CAs (*e.g.*, issuance and revocation of certificates) have become more important than ever. In this work, we comprehensively studied the security practices of CAs. We first have shown the current landscape of HTTPS phishing attacks—including the effectiveness of the attack. We then have revealed significant security concerns based on conflicts between the expected security practices of CAs and reality through the empirical experiments. In addition, we have observed the same conflicts and more security concerns in the wild by analyzing a longitudinal dataset of abusive certificates directly issued to adversaries.

## ACKNOWLEDGMENTS

We thank the anonymous referees for their constructive feedback. We also thank Adam Oest for his support in this study, and particularly for his contribution to the data collection and the analysis. The authors gratefully acknowledge the support of National Science Foundation (Grants No. CNS-1916499, CNS-1850392, CNS-1703644, CNS-1651661, and OAC-1908021), Defense Advanced Research Projects Agency (Grant No. HR001118C0060 and FA875019C 0003), National Research Foundation of Korea (Grant No. NRF-2020R1C1C1009031), and the Institute for Information & communications Technology Promotion (Grant No. 2017-0-00168). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

## REFERENCES

[1] 1987. Domain names - implementation and specification. RFC 1035. https://doi.org/10.17487/RFC1035
[2] (Accessed on 10/12/2020). GitHub - elceef/dnstwist: Domain name permutation engine for detecting homograph phishing attacks, typo squatting, and brand impersonation. https://github.com/elceef/dnstwist.
[3] (Accessed on 10/12/2020). RFC 2119: Key words for use in RFCs to Indicate Requirement Levels. https://tools.ietf.org/html/rfc2119.
[4] (Accessed on 10/14/2020). Alexa - Top sites. https://www.alexa.com/topsites.
[5] (Accessed on 10/14/2020). Cisco Umbrella 1 Million - Cisco Umbrella. https://umbrella.cisco.com/blog/cisco-umbrella-1-million.
[6] (Accessed on 10/15/2020). *Anti-Phishing Working Group.* https://apwg.org.
[7] (Accessed on 10/15/2020). *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates.* https://cabforum.org/baseline-requirements-documents/.
[8] (Accessed on 10/15/2020). *Certificate Revocation Lists: Certificates Revoked per Day.* https://isc.sans.edu/crls.html.
[9] (Accessed on 10/15/2020). *Comodo Certification Practice Statement.* https://sectigo.com/uploads/files/Sectigo-CPS-v5.1.1.pdf.
[10] (Accessed on 10/15/2020). *DigiCert: Certification Practices Statement—Version 5.4, September 29, 2020).* https://www.digicert.com/wp-content/uploads/2020/09/DigiCert-CPS-V.5.4.pdf.
[11] (Accessed on 10/15/2020). *DigiNotar SSL certificate hack amounts to cyberwar, says expert.* https://www.theguardian.com/technology/2011/sep/05/diginotar-certificate-hack-cyberwar.
[12] (Accessed on 10/15/2020). *GlobalSign Certification Practice Statement.* https://www.globalsign.com/en/repository/GlobalSign_CPS_v9.2_final.pdf.
[13] (Accessed on 10/15/2020). *Go Daddy Certificate Policy and Certification Practice Statement (CP/CPS) – Version 4.8, Sept. 30, 2020.* https://certs.godaddy.com/repository.
[14] (Accessed on 10/15/2020). *HEARTBLEED UPDATE (V3).* https://blogs.akamai.com/2014/04/heartbleed-update-v3.html.
[15] (Accessed on 10/15/2020). Internet Crime Complaint Center (IC3) | Cyber Actors Exploit 'Secure' Websites In Phishing Campaigns. https://www.ic3.gov/Media/Y2019/PSA190610.
[16] (Accessed on 10/15/2020). *Internet Security Research Group (ISRG) Certification Practice Statement—v2.9.* https://letsencrypt.org/documents/isrg-cps-v2.9/.
[17] (Accessed on 10/15/2020). Majestic Million - Majestic. https://majestic.com/reports/majestic-million.
[18] (Accessed on 10/15/2020). More Than Half of Phishing Sites Now Use HTTPS. https://info.phishlabs.com/blog/more-than-half-of-phishing-sites-use-https.
[19] (Accessed on 10/15/2020). Phishing Schemes Are Using HTTPS Encrypted Sites to Seem Legit | WIRED. https://www.wired.com/story/phishing-schemes-use-encrypted-sites-to-seem-legit/.
[20] (Accessed on 10/15/2020). *The Results of the CloudFlare Challenge.* https://blog.cloudflare.com/the-results-of-the-cloudflare-challenge.
[21] (Accessed on 10/15/2020). *Sectigo Certification Practice Statement—CPS Version 5.2.2, September 30, 2020.* https://sectigo.com/uploads/files/Sectigo-CPS-v5_2_2.pdf.
[22] (Accessed on 10/16/2020). VirusTotal. https://www.virustotal.com/gui/.
[23] (Accessed on 10/30.2020). *Anti-Phishing Working Group: APWG Trends Report Q2 2020.* https://docs.apwg.org/reports/apwg_trends_report_q2_2020.pdf.
[24] (accessed September 14, 2020). *APWG eCrime Exchange.* https://apwg.org/ecx/.
[25] Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, et al. 2019. Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security.* 2473–2487.
[26] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015).* Internet Society.
[27] R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten. 2019. *Automatic Certificate Management Environment (ACME).* RFC 8555. RFC Editor.

[28] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. 2008. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.* RFC 5280. RFC Editor. http://www.rfc-editor.org/rfc/rfc5280.txt http://www.rfc-editor.org/rfc/rfc5280.txt.

[29] Don Coppersmith. 1997. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology* 10, 4 (1997), 233–260.

[30] Zakir Durumeric, James Kasten, Michael Bailey, and J. Alex Halderman. 2013. Analysis of the HTTPS Certificate Ecosystem. In *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC '13)*. New York, NY, USA, 291–304.

[31] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, and J. Alex Halderman. 2014. The Matter of Heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference (IMC '14)*. 475–488.

[32] Adrienne Porter Felt, Robert W Reeder, Alex Ainslie, Helen Harris, Max Walker, Christopher Thompson, Mustafa Embre Acer, Elisabeth Morant, and Sunny Consolvo. 2016. Rethinking connection security indicators. In *12th Symposium on Usable Privacy and Security (SOUPS '16)*. 1–14.

[33] Google. (Accessed on 10/15/2020). Google Safe Browsing – Google Transparency Report. https://transparencyreport.google.com/safe-browsing/overview?hl=en.

[34] Tobias Holgers, David E. Watson, and Steven D. Gribble. 2006. Cutting Through the Confusion: A Measurement Study of Homograph Attacks. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference (ATEC '06)*. USENIX Association, Berkeley, CA, USA, 24–24.

[35] Doowon Kim, Bum Jun Kwon, Kristián Kozák, Christopher Gates, and Tudor Dumitras. 2018. The Broken Shield: Measuring Revocation Effectiveness in the Windows Code-Signing PKI. In *27th USENIX Security Symposium (USENIX Security 18)*. USENIX Association, Baltimore, MD, 851–868.

[36] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. ACM, New York, NY, USA, 569–586.

[37] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 569–586.

[38] Katharina Krombholz, Karoline Busse, Katharina Pfeffer, Matthew Smith, and Emanuel von Zezschwitz. 2019. "If HTTPS Were Secure, I Wouldn't Need 2FA"-End User and Administrator Mental Models of HTTPS. *IEEE Security & Privacy* (2019).

[39] Deepak Kumar, Zhengping Wang, Matthew Hyder, Joseph Dickinson, Gabrielle Beck, David Adrian, Joshua Mason, Zakir Durumeric, J Alex Halderman, and Michael Bailey. 2018. Tracking certificate misissuance in the wild. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 785–798.

[40] B. Laurie, A. Langley, and E. Kasper. 2013. *Certificate Transparency*. RFC 6962. RFC Editor.

[41] Yabing Liu, Will Tome, Liang Zhang, David Choffnes, Dave Levin, Bruce Maggs, Alan Mislove, Aaron Schulman, and Christo Wilson. 2015. An end-to-end measurement of certificate revocation in the web's PKI. In *Proceedings of the 2015 Internet Measurement Conference*. ACM, 183–196.

[42] M. Lochter and J. Merkle. 2010. *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation.* RFC 5639. RFC Editor.

[43] Zane Ma, Joshua Reynolds, Joseph Dickinson, Kaishen Wang, Taylor Judd, Joseph D Barnes, Joshua Mason, and Michael Bailey. 2019. The impact of secure transport protocols on phishing efficacy. In *12th USENIX Workshop on Cyber Security Experimentation and Test (CSET 19)*.

[44] Trend Micro. 2019. Security News: HTTPS Protocol Now Used in 58% of Phishing Websites. https://www.trendmicro.com/vinfo/hk-en/security/news/cybercrime-and-digital-threats/https-protocol-now-used-in-58-of-phishing-websites. (Accessed on 10/15/2020).

[45] Tyler Moore and Richard Clayton. 2007. Examining the impact of website take-down on phishing. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. 1–13.

[46] Nick Nikiforakis, Steven Van Acker, Wannes Meert, Lieven Desmet, Frank Piessens, and Wouter Joosen. 2013. Bitsquatting: Exploiting Bit-flips for Fun, or Profit?. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13)*. ACM, New York, NY, USA, 989–998.

[47] Adam Oest, Yeganeh Safaei, Adam Doupe, Gail-Joon Ahn, Brad Wardman, and Kevin Tyers. 2019. PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques against Browser Phishing Blacklists. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1344–1361.

[48] Adam Oest, Penghui Zhang, Brad Wardman, Eric Nunes, Jakub Burgis, Ali Zand, Kurt Thomas, Adam Doupé, and Gail-Joon Ahn. 2020. Sunrise to Sunset: Analyzing the End-to-end Life Cycle and Effectiveness of Phishing Attacks at Scale. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security 20)*.

[49] Federal Bureau of Investigation. (Accessed on 10/14/2020). 2019 Internet Crime Report. https://pdf.ic3.gov/2019_IC3Report.pdf.

[50] Peng Peng, Chao Xu, Luke Quinn, Hang Hu, Bimal Viswanath, and Gang Wang. 2019. What Happens After You Leak Your Password: Understanding Credential Sharing on Phishing Sites. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS '19)*. New York, NY, USA, 12.

[51] Richard Roberts, Yaelle Goldschlag, Rachel Walter, Taejoong Chung, Alan Mislove, and Dave Levin. 2019. You are who you appear to be: a longitudinal study of domain impersonation in TLS certificates. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2489–2504.

[52] Scott Ruoti, Tyler Monson, Justin Wu, Daniel Zappala, and Kent Seamons. 2017. Weighing context and trade-offs: How suburban adults selected their online security posture. In *13th Symposium on Usable Privacy and Security (SOUPS '17)*.

[53] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. 2013. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP.* RFC 6960. RFC Editor. http://www.rfc-editor.org/rfc/rfc6960.txt http://www.rfc-editor.org/rfc/rfc6960.txt.

[54] Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C Schmidt, and Matthias Wählisch. 2018. The rise of certificate transparency and its implications on the Internet ecosystem. In *Proceedings of the Internet Measurement Conference 2018*. 343–349.

[55] Christopher Thompson, Martin Shelton, Emily Stark, Maximilian Walker, Emily Schechter, and Adrienne Porter Felt. 2019. The web's identity crisis: understanding the effectiveness of website identity indicators. In *28th USENIX Security Symposium (USENIX Security 19)*. 1715–1732.

[56] Ke Tian, Steve T. K. Jan, Hang Hu, Danfeng Yao, and Gang Wang. 2018. Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild. In *Proceedings of the Internet Measurement Conference 2018 (IMC '18)*. New York, NY, USA, 429–442.

[57] Benjamin VanderSloot, Johanna Amann, Matthew Bernhard, Zakir Durumeric, Michael Bailey, and J Alex Halderman. [n.d.]. Towards a complete view of the certificate ecosystem. In *Proceedings of the 2016 Internet Measurement Conference (IMC '16)*.

[58] Michael J Wiener. 1990. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information theory* 36, 3 (1990), 553–558.

[59] Scott Yilek, Eric Rescorla, Hovav Shacham, Brandon Enright, and Stefan Savage. 2009. When Private Keys Are Public: Results from the 2008 Debian OpenSSL Vulnerability. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement (IMC '09)*. ACM, New York, NY, USA, 15–27.

[60] Liang Zhang, David Choffnes, Dave Levin, Tudor Dumitraş, Alan Mislove, Aaron Schulman, and Christo Wilson. 2014. Analysis of SSL certificate reissues and revocations in the wake of Heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 489–502.

[61] Penghui Zhang, Adam Oest, Haehyun Cho, RC Johnson, Brad Wardman, Shaown Sarker, Alexandros Kpravelos, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. 2021. CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. In *In Proceedings of the 42nd IEEE Symposium on Security and Privacy (Oakland)*. San Francisco, CA.

# APPENDIX

Table 6: **TLS Certificate Authority Market.** Unless the CAs claim their maximum revocation delay, the revocation delay specified in the CA/B Baseline Requirements are listed in this table.

| CA | Reseller | Price (DV) | Validity Period | Revocation | |
|---|---|---|---|---|---|
| | | | | Delay | Method |
| Let'sEncrypt | - | $0.00 | 90d | 1d [16] | E |
| cPanel | - | $0.00 | 90d | 1-5d [9] | E |
| CloudFlare | - | $0.00 | 6m | 1-5d [10] | W,E |
| COMODO | - | $99.00 | 1–2y | 1-5d [21] | E |
| | comodosslstore | $8.95 | 1–2y | 1-5d [21] | E |
| DigiCert | - | $218.00 | 1–2y | 1-5d [10] | E |
| RapidSSL | - | $59.00 | 1–2y | 1-5d [10] | W,E |
| | thesslstore | $17.95 | 1–2y | 1-5d [10] | W,E |
| GeoTrust | - | $149.00 | 1–2y | 1-5d [10] | W,E |
| | cheapsslsecurity | $44.95 | 1–2y | 1-5d [10] | W,E |
| GlobalSign | - | $249.00 | 1–2y | 1-5d [12] | W,E |
| Go Daddy | - | $79.99 | 1–2y | 1-5d [13] | W,E |

* W: web form, E: email

Table 7: **CT Logs of the issued high-risk domain certificates.** They are issued with the suspicious domain name (*pyp-al.com*) that had been already used for a real phishing attack.

| crt.sh ID | Logged At | Issuer |
|---|---|---|
| 1950796162 | 2019-10-02 | cPanel |
| 1950666605 | 2019-10-02 | Let's Encrypt |
| 1950582955 | 2019-10-02 | GlobalSign |
| 1949985542 | 2019-10-02 | GeoTrust *(re-seller:* cheapsslsecurity*)* |
| 1949774788 | 2019-10-02 | GeoTrust |
| 1949917389 | 2019-10-02 | RapidSSL *(re-seller:* thesslstore*)* |
| 1947266413 | 2019-10-02 | RapidSSL |
| 1949842149 | 2019-10-02 | Comodo *(re-seller:* comodosslstore*)* |
| 1947421234 | 2019-10-02 | Comodo |
| 1947184524 | 2019-10-02 | Go Daddy |

Table 8: **Top-10 Issuers (CAs) including CloudFlare.** Revocation status is checked on Sep. 22, 2020. We are unable to check the revocation status of 23 certificates due to OCSP errors.

| Cat. | Issuer (CA) | Total | Status | | |
|---|---|---|---|---|---|
| | | | Good | Expired | Revoked |
| A/C | COMODO | 2,259,250 | 10,044 | 2,249,161 | 45 |
| A | Let's Encrypt | 1,951,721 | 130,708 | 1,820,996 | 17 |
| A | cPanel | 1,197,283 | 57,710 | 1,139,566 | 7 |
| C | GlobalSign | 120,855 | 17,453 | 103,252 | 150 |
| A | CloudFlare | 58,136 | 36,563 | 21,573 | 0 |
| C | Go Daddy | 34,175 | 7,458 | 24,937 | 1,780 |
| C | Sectigo | 30,233 | 11,355 | 18,617 | 261 |
| C | DigiCert | 19,802 | 8715 | 11,003 | 84 |
| C | RapidSSL | 9,772 | 1,300 | 8,465 | 7 |
| C | GeoTrust | 5,857 | 855 | 4,995 | 7 |
| | Etc. | 52,254 | 11,477 | 44,528 | 537 |
| | **Total** | 5,739,338 | 293,638 | 5,442,805 | 2,895 |

A: free, automated CA, C: commercial CA.

Table 9: **CPS/CP revocation statements by CAs.**

| CA | CPS/CP Revocation Statement |
|---|---|
| Comodo | "The Subscriber has used the Certificate contrary to law, rule or regulation, or Sectigo reasonably believes that the Subscriber is using the Certificate, directly or indirectly, to engage in illegal or fraudulent activity." |
| Let's Encrypt | "Each Subscriber acknowledges and accepts that ISRG is entitled to revoke Subscriber's ISRG certificates immediately if the Subscriber violates the terms of the Subscriber Agreement or if ISRG discovers that any of Subscriber's ISRG certificates are being used to enable criminal activities such as phishing attacks, fraud, or the distribution of malware." |
| Go Daddy | "Within 24 hours after receiving a Certificate Problem Report, ... provide a preliminary report on its findings to both the Subscriber and the entity who filed the Certificate Problem Report." |