A Residential Client-side Perspective on SSL Certificates

Edward Oakes*, Jeffery Kline†, Aaron Cahn‡, Keith Funkhouser§, Paul Barford¶
*University of California, Berkeley. Email: eoakes@berkeley.edu
†Hitwise. Email: jkline@hitwise.com
‡Comscore. Email: acahn@comscore.com
§Comscore (now at Google). Email: keithfunkhouser@gmail.com
¶University of Wisconsin, Madison. Email: pb@cs.wisc.edu

***Abstract*—SSL certificates are a core component of the public key infrastructure that underpins encrypted communication in the Internet. In this paper, we report the results of a longitudinal study of the characteristics of SSL certificate chains presented to clients during secure web (HTTPS) connection setup. Our data set consists of 23B SSL certificate chains collected from a global panel consisting of over 2M residential client machines over a period of 6 months. The data informing our analyses provide perspective on the entire chain of trust, including root certificates, across a wide distribution of client machines. We identify over 35M unique certificate chains with diverse relationships at all levels of the PKI hierarchy. We report on the characteristics of valid certificates, which make up 99.7% of the total corpus. We also examine invalid certificate chains, finding that 93% of them contain an untrusted root certificate and we find they have shorter average chain length than their valid counterparts. Finally, we examine two unintended but prevalent behaviors in our data: the deprecation of root certificates and secure traffic interception. Our results support aspects of prior, scan-based studies on certificate characteristics but contradict other findings, highlighting the importance of the residential client-side perspective.**

I. Introduction

Secure communication in the modern web via HTTPS is increasingly prevalent, and is facilitated by Transport Layer Security (TLS) and its precursor Secure Socket Layer (SSL).[1] SSL is a cryptographic protocol [1] designed to allow two communicating applications to communicate privately and to guarantee the integrity of transmitted messages. The protocol relies on a secure public key infrastructure (PKI), which relies on digital certificates that prove ownership of a public key associated with a domain name. While the SSL protocol is precisely defined and relatively stable, the larger ecosystem is complex and dynamic. In spite of on-going efforts to increase transparency, the state of the general ecosystem is only partially understood [2], [3], [4].

To be effective at securing online communication, the PKI must be actively maintained by package developers and routinely updated by the clients where it is deployed. While client-side updates are automated, PKI maintenance is currently a manual endeavor that demands familiarity with software project policies and industry best practices. Maintainers of major operating systems, browsers and other applications that include a repository of trusted root certificates must regularly, and subjectively assess the trustworthiness of certificate signing authorities. The true efficacy of the PKI at securing real-world online communication therefore cannot be assessed from the narrow perspective of whether certificate chains adhere to technical specifications. Rather, assessing the efficacy of the PKI requires a perspective that can reveal how this infrastructure is used in organic traffic by a broad set of clients.

The study of SSL certificate chains in general presents several challenges. First is the enormous scale, complexity, diversity, and dynamics of the web itself. Second, as prior studies have identified, the SSL ecosystem is also large, diverse, and dynamic. Third, to be relevant, the perspective of key participants in web transactions, *i.e.*, clients, servers, and intermediaries, should be represented in the measurements.

Standard methods for studying SSL include repeated scans of the IPv4 address space in search of servers that respond to connection requests on port 443 [4], [5], [6] or passively observing networks [7]. These studies have highlighted a range of critical characteristics of valid and invalid certificates as well as unexpected behaviors. Other surveys have explored the landscape from the perspective of reports from the web browser [8]. A recent unifying view of the SSL system was described in [9]. While the authors consider multiple perspectives, missing from their view is the *client-side perspective*. Finally, Flash ads have been used to study the SSL ecosystem [5], [10]. However, Flash has since been deprecated in favor of other technologies by major industry groups, and as a result, future studies will be unable to use this technique.

Our study is based on a compelling corpus of data: a set of over 23 billion SSL certificate chains collected during daily web browsing by a 2 million person world-wide residential user panel over a 6-month period. We identify over 35 million unique chains in this data set with diverse relationships between root, intermediate, and leaf certificates. To the best of our knowledge, this represents the largest user-generated SSL data set considered in a research study. Using this data, we revisit findings from prior studies and report new findings that expand the perspective of the current SSL ecosystem.

We begin by examining the general characteristics of the certificates in our data set. We then compare and contrast

[1] We will adopt the common practice of referring to both TLS and SSL as "SSL" for the remainder of this paper.

978-3-903176-17-1 / © 2019 IFIP

the properties of certificate chains that pass or fail a standard validation process. Our analysis focuses on certificate characteristics such as chain length, certificate type diversity, key and signature algorithms, validity periods, and certificate constraints. We find that the vast majority of certificate chains utilized in the SSL ecosystem are valid, they are associated with large and prominent domains, and they rely on trusted intermediaries and root authorities that follow best practices. In contrast, we observe that invalid certificate chains generally have shorter length than their valid counterparts, they generally fail validation due to a non-standard root certificate anchor and, surprisingly, a *majority* (73%) of the distinct chains that we observed were invalid. We find that the community efforts to end the use of SHA1 hashing functions and 1024-bit certificates have made progress but are still incomplete. Finally, we observe that certificates found unsuitable for deployment by major software projects due to policy violations can linger in the ecosystem for years, resulting in substandard certificates playing a significant role in securing online traffic.

II. SSL Overview

Clients, such as web browsers, verify server identity and establish a trusted communication channel by leveraging the SSL protocol, a comprehensive overview of which is provided in [11]. Trust is represented by digitally signed certificates and is established hierarchically. The basis of all trusted communication for clients rests upon a small and curated set of root certificates from trusted certificate authorities (CAs). These certificates have the highest level of trust and may validate intermediate certificates or terminal leaf certificates. Intermediate certificates may in turn validate additional intermediate certificates and terminal leaf certificates. Every valid leaf certificate is associated with a chain of trust that terminates at a trusted root certificate. A set of root and intermediate certificates distributed with operating systems and web browsers is used by clients to validate chains of trust.

Ideally, the set of trusted root and intermediate certificates deployed on client machines would be standardized and up to date. However, in practice this certificate store varies over time and by platform. A client's local repository of trusted certificates is updated via routine operating system updates and with the installation of new software applications. Updates can also be triggered by malicious activity. As a result, the chain of trust used by one client to validate a certificate will not, in general, match the chain used by another. In fact, a certificate may have multiple valid chains of trust on the same client machine.

Certificates themselves are documents that contain descriptive fields, such as a *common name* (CN), *subject alternative name* (SAN), the name and domain of its owner, the name and domain of its issuer, a valid time window, and a URL pointing to a certificate revocation list (CRL) that is used to determine whether the certificate has been revoked by its CA. RFC 5280 [11] is unequivocal that CAs have the responsibility to verify the information of entities to which they issue certificates. To adhere to this principal and still allow CAs to issue certificates at scale without succumbing to administrative overhead, the IETF developed the Automatic Certificate Management Environment (ACME) protocol [12]. This protocol enables CAs to confirm that an applicant for a certificate controls a particular domain, without human intervention. The decentralized architecture of the PKI means that no global certificate-domain pairing registry exists, so a domain may be secured by many certificates.

Although the PKI is simple in concept, it relies on the ability of a diverse set of entities to maintain it and follow general security best-practices, *e.g.,* maintaining the secrecy of their private key. In practice, the complexity of this ecosystem results in unintended uses and consequences of the protocol. Therefore, we posit that repeated, representative measurement studies are essential to the continued maintenance of the PKI.

III. Methodology

A. Client-side Data Collection

We obtain our SSL certificate chain data from the Comscore global desktop user panel. The panel is made up of over 2 million residential participants who elect to install web monitoring software in exchange for benefits such as cash awards, antivirus software, and online credits. Participation is voluntary and requires informed consent. Data are handled in accordance with Comscore's privacy policies [13].

The client-side panel software enables Comscore to monitor and collect data transmitted by panelists while using internet-enabled applications. Each time a panelist machine validates a certificate chain as part of an SSL handshake, the full certificate chain is recorded by the client-side panel software and sent to Comscore storage servers. An unusual feature of our data is that the certificate chain includes intermediate and root certificates that may not have been transmitted over the network but were appended by the client.

The data set provided by Comscore consists of certificate chains collected from panelists during the 185-day period from July 12, 2017 to January 12, 2018. Table I provides an overview of the corpus of certificate chains. While the certificates collected may include personally identifiable information (PII) such as email addresses or software serial numbers included in custom certificates, our analysis consists primarily of high level aggregations.

Panel collection servers typically ingest tens of billions of records daily, and over the 6-month observation period, panelists issued requests to 9,310 of the Alexa Top 10,000 Sites, including all 100 of the top 100 sites.

B. Data Processing

Each certificate chain is validated using the open-source OpenSSL tool version 1.0.1 [14] with the trusted CA store shipped with Linux CentOS 7. After labeling chains as *valid* or *invalid*, we decode and parse fields from each certificate using the Python open-source cryptography package [15]. We define a *valid* certificate chain as one that has a well-formed chain of trust with verified signatures from the leaf certificate

TABLE I

A summary of our data. A one-to-one correspondence exists between root certificates and certificate chains. The count of leaf certificates is less than the count of certificate chains because the certificate in a chain of length 1 is categorized as a root (not a leaf).

Record Type	Unique Count	Total Count
All Certificate Chains	35,072,572	22,750,189,641
Valid Certificate Chains	9,655,031	22,680,255,554
Root Certificates	166,670	22,750,189,641
Intermediate Certificates	3,512,742	26,509,138,959
Leaf Certificates	31,574,086	22,750,189,252

TABLE II

The causes of certificate chain verification failure. "Empty Validity Period" indicates that the intersection of the validity periods of all certificates in a chain is empty. We do not define "expired" for unique chains because this set is constructed without considering observed timestamps.

Failure Reason	Unique Chains	Total Chains
Untrusted Root Certificate	25,263,314	65,104,190
Missing Issuer Certificate	151,285	3,491,725
Expired	N/A	1,316,413
Signature Failure	2,515	10,538
Improperly Formatted	368	10,513
Empty Validity Period	56	629
Self-signed Certificate	10	79

to a publicly trusted root CA and was observed during the validity period of all certificates in the chain. Any certificate chain that does not meet these conditions is considered *invalid*. We recognize this definition will label some well-formed and non-malicious certificate chains that are trusted by clients as invalid. For example, chains that contain a client-trusted custom root CA, such as those installed by a secure proxy, would all be labeled invalid. A chain, in this circumstance, we will call *strictly locally valid*. The total deduplicated corpus of certificates chains and metadata is 926 GB. Validating the chains and decoding the certificates required about 120 hours of computation time on an Apache Spark cluster that consists of several hundred nodes.

IV. Characteristics of Client-side SSL Certificates

In this section, we report the characteristics of certificate chains observed in our data. Where appropriate, we compare and contrast our observations with prior work.

A. Certificate Chain Validity

Table I provides a snapshot of our data at the end of our collection period. Of the 22.8B chains observed, 99.7% were labeled valid. In contrast to this, just 9.7M, or 27.5%, of the 35M distinct chains observed were labeled valid. To supplement this view, Figure 1 shows the aggregated count of unique certificate chains observed over the six months of observations as well as the least-square linear fit of each time series. The slope of the fit, whose units are *newly-observed chains-per-day*, quantifies the arrival rate of new distinct certificate chains.

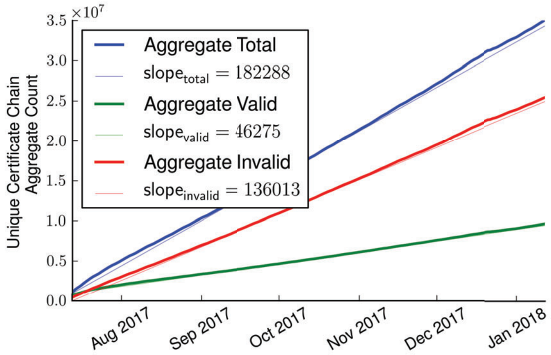

Fig. 1. The aggregate count of unique total, valid, and invalid certificate chains collected over time. Also shown is a linear fit to each time series, with the slope of each fit indicated. The units of the slope are *newly-observed chains-per-day*. In each case the r^2-value of the linear fit exceeds 0.99.

Valid certificate chains constitute a minority of the unique certificate chain population yet they are the vast majority of total chains that we observe. This is a reflection of the characteristic of organic web traffic: most traffic by volume flows toward a small number of prominent domains. Popular HTTPS-enabled web sites and online services, in order to function, must have properly configured and stable SSL deployments. As a result, a few popular web servers perform vastly more SSL handshakes than those of less popular sites and services. This observation has a natural dual: unmaintained or seldom used servers that accept SSL traffic on the Internet are less likely to serve properly configured SSL certificate chains than popular entities that are incentivized to do so.

This argument is supported by the observation that 90% of valid certificate chains collected by volume are comprised of only 6.5K unique certificate chains, while the 90% mark for invalid certificates by volume is comprised of 15M unique chains. This difference indicates that there is a large disparity in popularity between services that provide valid and invalid certificate chains, and there is a long tail of root and intermediate certificate authorities used by invalid certificate chains. In comparison to prior studies [4], [16] based on port 443 scans of the IPv4 address space, we find that users encounter a significantly smaller proportion of improperly configured, expired, and not yet valid certificates.

Examining the certificate chains that are labeled invalid in our data, Table II shows that the most common cause of validation error is the presence of an untrusted root certificate. We observe that when a root certificate in our data is untrusted, it is usually associated uniquely with a single machine. This is consistent with certificates generated during the installation process of widely-deployed antivirus software and content filters [17]. Requests that originate from a device configured to use such a custom certificate will present certificate chains that are unique to the device. As a result, traffic from such

TABLE III
The distribution of chain lengths for valid and invalid certificate chains. Valid certificate chains have median length 3 while invalid certificate chains have median length 2.

Length	Unique Valid	Total Valid	Unique Invalid	Total Invalid
1	0	0	11,353	27,806
2	316,350	273,435,542	22,798,458	51,973,767
3	7,086,677	19,360,646,830	1,623,554	16,895,822
4	2,141,143	2,443,150,955	952,622	2,526,018
5	109,403	688,444,324	31,302	142,812
6	1,255	20,200	243	1,047
7	91	19,969	5	84
8	41	86	7	51
≥ 9	71	9,364	4	859

TABLE IV
Certificate authority organizations ranked by the number of distinct trusted root certificates that they issued. Not all of these organizations are independent of each other.

Organization	Unique Root Certificates	
Symantec Corporation	24,872	(30.8%)
GlobalSign nv-sa	23,212	(28.7%)
GeoTrust Inc.	14,407	(17.8%)
Entrust, Inc.	6,848	(8.5%)
thawte, Inc.	5,791	(7.2%)
Remaining 193	3,167	(3.9%)

TABLE V
Certificate authority organizations ranked by the number of distinct trusted intermediate certificates that they issued.

Organization	Unique Intermediate Certificates	
COMODO CA Limited	74,258	(25.4%)
Let's Encrypt	61,909	(21.2%)
DigiCert Inc	27,685	(9.5%)
GoDaddy.com, Inc.	24,137	(8.3%)
GeoTrust Inc.	20,766	(7.1%)
Remaining 213	28,490	(9.8%)

devices will contribute disproportionately to the total count of unique chains that we observe. We discuss the presence of such certificate chains in our data in Section V.

B. Certificate Chain Length

An SSL certificate chain is comprised of a series of authorizations from root to intermediate to leaf certificates. *Chain length* is defined as the total number of signed certificates appearing in a chain. This is an important property of certificate chains for two reasons. First, longer chains decrease performance [18]. Second, the attack surface of a chain is proportional to the length of the chain. To bound this risk, SSL certificates that represent certificate authorities may possess an optional *path length* constraint. If present, this constraint is a nonnegative integer that indicates the maximum number of signing certificates that can follow the certificate in the chain and constitute a valid chain of trust [11]. If this constraint is unspecified, the chain may be of arbitrary length. Table III shows the distribution of chain lengths in our data by validity for both the total volume of chains and unique chains observed.

We find that the majority of root certificates (91%) had no path length constraint listed. In contrast, most valid intermediate certificates (79%) had a path length constraint of 0, indicating that they may only sign leaf certificates. These considerable majorities reflect the common practice of certificate authority organizations using root certificates to issue themselves an intermediate certificate, and using that certificate to sign leaf certificates. Nearly all of the remaining intermediate leaf certificates (20%) had no path length constraint, indicating that they may be followed by a chain of any number of intermediate certificates. In total, 99% of intermediate certificates preset in valid chains either have no path length constraint or have it set to 0.

C. Diversity of PKI Entities

We observe 294 issuing organizations that issued trusted root and intermediate certificates. Of these organizations, 203 issued root certificates and 223 issued intermediate certificates. Furthermore, the top ten organizations issued 89% of the trusted signing certificates in our data. This result aligns with previous findings that a handful of key players control most of the signing authority in the SSL ecosystem [19]. Tables IV and V list the top issuing organizations for root and intermediate certificates in our data, respectively.

D. Key and Signature Algorithms

Due to ever-increasing computational capabilities and newly-discovered vulnerabilities, the robustness of widely-deployed cryptographic algorithms needs to be checked on an on-going basis.

The current SSL key algorithm recommendation by NIST is to use RSA (2048 or 3072 bits) with SHA-256, ECDSA (Curve P-256) with SHA-256, or ECDSA (Curve P-384) with SHA-384 [20]. Table VI shows the distribution of the cryptographic key types observed in our data. We find that the vast majority of leaf certificates use 2048-bit RSA keys (77%) or 256-bit ECDSA keys (22%), in line with the NIST recommendations. Only a tiny fraction (0.66%) of leaf certificates use any other type of key. RSA is the dominant algorithm, with roughly 96% coverage, as can be seen in Table VII. Among the top 1,000 most popular root, intermediate, and leaf certificates seen in our data, we find that the most popular key algorithm by an order of magnitude is RSA-2048 (Table VIII), indicating that, as expected, popular entities are following NIST recommendations. The distribution of key algorithm popularity is very similar between root and intermediate certificates, while popular leaf certificates more frequently use ECDSA (256-bit).

A practical SHA1 collision attack was demonstrated in 2017, indicating that SHA1 is too weak to be considered secure [21]. The wider community began deprecating SHA1 for use in critical security infrastructure with the goal of fully eliminating SHA1 from new certificates by 2016 [22], [23], [24]. We also note that the RSA-768 factoring challenge was solved in 2009 by Kleinjung *et al.* [25].

Contrary to prior findings [19], [4], we find that only a very small fraction of valid leaf certificates use RSA with the SHA1 hashing algorithm. This successful widespread deprecation follows a NIST recommendation in 2015 and major browsers no longer trusting certificates that use it [26], [27], [28], [29]. We note that although the proportion is small, a substantial *number* of certificates (over 157K) persist in using SHA1.

In a similar vein, browser distributions [30], [31] and CA's [32], [33] began deprecating the use of 1024-bit certificates over the looming vulnerability to attack around 2013. This action was in response to the 2011-era NIST recommendation to transition to stronger encryption schemes [34]. In our data, only 0.06% of certificates using 1024-bit RSA keys exist as part of a valid chain.

V. Unintended Behaviors

We now explore two unintended behaviors related to the PKI that are observed in our data: SSL traffic interception [35] and deprecated certificates (defined below). While intercepted SSL traffic is a known phenomenon, we believe this is the first report on the prevalence of deprecated certificates. Although deprecation is not an explicitly defined designation of the PKI, we show it is a prevalent, well-defined feature.

A. Deprecated Certificates

We label a certificate *deprecated* if it is valid at the time of processing and has been identified for removal by the maintainers of at least one highly reputable CA certificate repository. While not all software projects openly disclose trusted CA repository maintenance, several projects maintain and distribute trusted certificate repositories, and also post archival information about historical changes to their trusted root repositories. The projects we select are: Debian Linux [36], the Fedora Project [37], Mozilla [38], Chrome [39] and Microsoft [40]. In late 2016, the Fedora project began using an unmodified version of the trusted store published by Mozilla. Each project maintains its own repository with transparency and according to a publicly-posted code of principles. We also rely on the online certificate catalog hosted at `https://www.crt.sh` to inform our assignment of certificate revocation status.

As trust is an inherently nontechnical concept, each of the selected projects relies on manual curation of its trusted root certificate store. We identify deprecated certificates from these projects via manual search over CA certificate package maintenance forums and through a review of publicly-accessible archival data [41], [42].

Each project has its own policies in place for root certificate validation and removal, typically requiring a subjective assessment of CA compliance with software project policy. Common reasons for deprecation include the widespread deprecation of SHA1 [28], [29], [24] and compliance issues that arise from a CA retiring, but not revoking, certificates [43], [30]. Although trusted root certificate stores are maintained by independent groups, some projects coordinate loosely. However, inconsistent policies governing CA root certificate removal across projects and logistical complexities suggest that a more formal synchronization process is impractical. To our knowledge, there is no public catalog of these deprecated certificates. This lack of standardization makes automated discovery of deprecated certificates a significant challenge, so we limit our analysis to manually-identified cases.

Turning to the presence of this feature in our data, we identify a total of 69 deprecated certificates via our process. As the identification of these certificates is manual, this number does not purport to be comprehensive, and there are undoubtedly more certificates which fit this profile in our data. For perspective on the significance of this number, the Microsoft trusted CA root repository, dated December 2017, contains 360 certificates. In our data, we observe 24 deprecated certificates in active use between July and December 2017.

The longer such certificates are in use, the more of a security threat they pose, so this trend has a negative impact on the PKI as a whole. Deprecated certificates are, by definition, widely deployed at some point in their valid existence. If a CA retires but does not revoke a certificate, then all chains that validate against this root may be secured against substandard or vulnerable trusted roots. As a single certificate may possess several valid chains of trust on a single machine, we observe that traffic to search engines, social media and other high-value domains are secured using the deprecated certificates that we identified. This holds even if domain administrators employ best practices to ensure the security of visitors to their online service.

B. Intercepted Certificate Chains and Malicious Behavior

We define an *intercepted chain* as a certificate chain whose root certificate has never been distributed as part of a trusted root store. Intercepted chains generally exist so that a third-party is able to decrypt secure traffic. The kinds of activities and software that we observe associated with intercepted certificates include virus filters, content filters, software development tools, commercial research, and malicious intent. At a more technical level, intercepted certificates occur because many implementations of SSL are relatively straightforward to circumvent. The primary exception to this is traffic secured by an application that uses pinned certificates [44], [45]. A sketch of SSL traffic interception in practice is as follows:

1) A custom root certificate is installed in a trusted location on the client machine.
2) A proxy is configured to relay client traffic.
3) When the client attempts to establish a secure connection to an external server, the proxy dynamically signs certificates using its own CA. The client then establishes a secure connection to the proxy rather than the external server. The root certificate in the newly-generated chain of trust is the custom root certificate of the proxy.

TABLE VI
THE TOP KEY ALGORITHMS LISTED ACCORDING TO THE NUMBER OF TRUSTED LEAF CERTIFICATES OBSERVED. THE TWO MOST PREVALENT ACCOUNT FOR OVER 99% OF ALL VALID CERTIFICATES OBSERVED. RSA 1024 WAS GENERALLY TARGETED TO BE PHASED OUT BY 2012.

Key Algorithm	Valid leaf Certificates	Total Leaf Certificates	Percent of Certificates Valid	Percent of Valid Certificates
RSA (2048-bit)	17,432,945,097	17,470,304,302	99.79%	76.86%
ECDSA (256-bit)	5,096,433,902	5,098,375,701	99.96%	22.47%
RSA (4096-bit)	137,111,745	139,462,361	98.31%	0.60%
ECDSA (384-bit)	12,525,978	12,526,963	99.99%	0.06%
RSA (3072-bit)	488,625	525,657	92.96%	0.00%
. . .				
RSA (1024-bit)	16,814	28,056,718	0.06%	0.00%

TABLE VII
THE TOP SIGNATURE ALGORITHMS RANKED BY THE NUMBER OF TRUSTED LEAF CERTIFICATES OBSERVED.

Signature Algorithm	Valid Leaf Certificates	Total Leaf Certificates	Percent of Certificates Valid	Percent of Valid Certificates
SHA-256 with RSA	21,629,065,116	21,693,936,696	99.70%	95.37%
SHA-256 with ECDSA	969,688,996	969,707,077	$> 99.99\%$	4.28%
SHA-512 with RSA	66,647,282	68,679,795	97.04%	0.29%
SHA-384 with RSA	14,433,491	14,442,609	99.94%	0.06%
SHA-1 with RSA	405,697	3,331,455	12.18%	0.00%

TABLE VIII
THE COUNT OF THE TOP FIVE KEY ALGORITHMS IN THE MOST OBSERVED 1,000 CERTIFICATES IN OUR DATA SET, RANKED BY THE NUMBER OF THOSE CERTIFICATES THAT USED THEM.

Key Algorithm	Root	Intermediate	Leaf
RSA (2048-bit)	872	880	854
ECDSA (256-bit)	76	78	137
RSA (4096-bit)	43	37	7
RSA (1024-bit)	6	1	0
ECDSA (384-bit)	3	4	2

4) The proxy establishes a legitimate secure connection to the server. The proxy may decrypt or modify traffic while relaying it between client and server.

Several online tutorials and software tools are available to guide the non-expert in configuring a process that can decrypt SSL traffic [46], [47], [48].

We now turn to certificates involved in interception whose purpose appears malicious. The label *likely malicious* is assigned to a certificate that 1) is the trusted anchor of an intercepted chain and 2) the certificate violates at least one of the following principles:

1) Transparency: Does the certificate contain contact information? Does the certificate identify its owner or purpose? For example, if the certificate is intended for benign interception, is this clearly indicated or easy to discover?
2) Conformance: Are the various attributes of the certificate created in accordance with common practice? For example, does the certificate's validity period exceed 100 years?

TABLE IX
MALICIOUS CERTIFICATES. A CERTIFICATE CREATOR MAY POPULATE THE ORGANIZATIONAL UNIT FIELD WITH AN ARBITRARY STRING, SUCH AS A WELL-KNOWN CA. WE HAVE BEEN UNABLE TO IDENTIFY THE TRUE OWNER OF “GLOBALSIGNATURE CERTIFICATES CA 2”.

# Chains	Subject Organizational Unit	Comment
24811	GlobalSignature Certificates CA 2	exp. 2056, mainly Chinese lang. domains
989	f53bd78a9cf13079 2	exp. 2057
977	VeriSign Trust Network	exp. 2115, Issued in 2015 with RSA 1 1024-bit
607	thawte 2	Issuer is ”C=EN, CN=thawte 2”
546	8683057bcb648b1f 2	exp. 2057

The certificates listed in Table IX are labeled likely malicious. They were used to secure traffic to a variety of online content including search engines, financial services, and social media. One certificate in this table, “GlobalSignature Certificates CA 2” is observed mainly securing traffic to Chinese-language domains. We have not been able to identify this CA, nor does it appear in searches of the certificate repository hosted at `crt.sh`.

Our data, in addition to logging all HTTP(S) communication that a machine engages in, also records the name of the process that initiated each request and whether the communication occurred securely or not. Additionally, Comscore maintains a list of process names that consistently engage in malicious activity. Process names are added to this list after empirical observation and manual forensic review assess whether the primary purpose of the process is malicious. This list is maintained and updated on an ongoing basis.

The malicious process list is expected to identify a proper subset of all malicious traffic, *i.e.,* it provides a partial view of the total volume of malicious traffic. Table X offers a high level description of the observed use of HTTP and HTTPS by malicious processes during one day in January, 2018. From this table it is evident that malicious processes do initiate secure requests, though the bulk of malicious web requests are insecure. This table also summarizes the use of HTTP and HTTPS when malware processes contact URLs that appear to be user or administrative login pages. The vast majority of such requests occur using HTTP. Of the requests identified as login attempts, the top response codes were 404-Not Found, 200-OK, and 301-Moved Permanently with volumes 418K, 246K, and 237K, respectively. The fraction of 200-OK responses from pages that request user credentials, about 25%, is surprisingly large.

TABLE X
HTTP(S) RECORDS OBSERVED ON 2018/01/10. MOST NON-SECURE REQUESTS BY MALICIOUS PROCESSES APPEAR TO BE LOGIN ATTEMPTS.

	Malicious login requests	All malicious requests	All other requests
HTTP	1,162,790	7,349,401	679,197,120
HTTPS	3,993	1,098,899	2,273,993,370
$\frac{\text{HTTP}}{\text{HTTPS}}$	291.2	6.69	0.30

The prevalence of secure traffic interception in our data suggests that it is widespread from the client perspective. Most intercepted traffic we observe is driven by antivirus software and content filter services. In order to function properly, these services must have unrestricted access to all traffic that is transmitted to or received by a machine. More broadly, the widespread deployment of pinned certificates and custom root certificates by a diverse range of applications and web browsers shows that certificate management is an exceedingly complex space. This is significant because RFC 5280 [11] prioritized “the development of certificate management systems,” among other things. Further study of this behavior and its impact on the trust hierarchy of the PKI is imperative to ensuring that the needs of the community are addressed.

VI. Related Work

Several prior studies of the SSL certificate ecosystem have been based on scans of port 443 across the entire IPv4 address space [19], [4]. These studies provide an important baseline for understanding valid SSL certificates that are accepted by standard browsers and follow best practices. In contrast, Chung *et al.* [6] use an extensive data set gathered by scanning to investigate SSL certificates that are identified as invalid. Kotzias *et al.* [49] combines perspectives of the ICSI Notary [50], which passively collects metadata about SSL/TLS connections from several universities and research networks and the Censys [51], which performs periodic active scans of the IPv4 space. While these studies inform our work and we reassess several of their key findings, we argue that the perspective offered by scan-based studies is not representative of how users experience SSL, thus client-based studies significantly augment our understanding of SSL in practice.

Huang *et al.* [5] implemented a Flash-based applet to collect SSL certificates from over 3 million client connections to Facebook’s website. They found that 0.2% of the SSL connections used forged certificates from antivirus software, organization-scale content filters, and malware. A survey of certificate errors that are reported by Chrome is undertaken in [52]. Our findings complement the results in these prior studies by highlighting how users experience and interact with the SSL ecosystem and also extend these results by drawing attention to practical aspects of PKI maintenance.

Holz *et al.* [16] conducted a large study on the security of SSL deployments for email and chat infrastructures. In their work they use two sources of data: active Internet-wide scans and passive monitoring of university campus traffic. Gasser *et al.* [53] use the Certificate Transparency (CT) logs to report on adherance to industry baseline requirements of TLS. Finally, Heniger *et al.* [54] find, using a network survey, that vulnerable host and ssh keys are surprisingly common due to weaknesses in standard random number generators.

VII. Summary and Future Work

In this work, we present a measurement study to illuminate the characteristics of the SSL ecosystem in practice, collecting and analyzing certificate chains presented during user-driven web browsing. Our technique differs significantly from previous work in the literature by virtue of our data corpus: we do not rely on Internet-wide port scans, which do not take into account browsing behaviors in the web. While some of our results confirm those from prior studies, our findings highlight that over 99% of the certificates *used* in SSL exchanges are valid even though the majority of SSL certificates *available* are invalid as reported in previous work [6].

Although the PKI is decentralized by design, our analysis reaffirms previous results that the hierarchy is skewed to rely heavily on a few entities. One outcome of this insight is that root certificates, which have the most authority in the PKI, are subject to extremely high security standards and are audited to ensure that they are met. However, our analysis has shown that the most popular intermediate signing certificates present in our data set have a similarly large presence.

Over the last few decades since the invention of SSL, the Internet landscape has shifted significantly due to increased user demands, shifts in content consumption, and the rise of the Internet advertising industry, among other things. Due to this high prevalence, examining the efficacy of SSL in this high-impact space and its implications for user experience and privacy is crucial. We leave this exploration to future work.

As with any Internet protocol, the SSL ecosystem is diverse and constantly evolving. Continual measurement and reevaluation of the protocol and its ability to address user demands is of utmost importance. This is highlighted by our exposure of the prevalence of informal certificate deprecation and secure traffic interception. These unintended behaviors indicate that certificate authorities and end users have both been left with demands that are incompletely fulfilled by the SSL protocol.

References

[1] T. Dierks and E. Rescorla, "RFC 5246: The Transport Layer Security (TLS) Protocol," Network Working Group, Tech. Rep., August 2008.

[2] J. Gustafsson, G. Overier, M. F. Arlitt, and N. Carlsson, "A First Look at the CT Landscape: Certificate Transparency Logs in Practice," in *PAM*, 2017.

[3] Comodo, "Comodo Launches New Digital Certificate Searchable Web Site," June 2015. [Online]. Available: https://www.comodo.com/news/press_releases/2015/06/comodo-launches-new-certificate-transparency-search-web-site.html

[4] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS Certificate Ecosystem," in *Proceedings of the ACM Internet Measurement Conference*, 2013.

[5] L. S. Huang, A. Rice, E. Ellingsen, and C. Jackson, "Analyzing Forged SSL Certificates in the Wild," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2014.

[6] T. Chung, Y. Liu, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "Measuring and Applying Invalid SSL Certificates: The Silent Majority," in *Proc. of the Internet Measurement Conference 2016*.

[7] D. Akhawe, B. Amann, M. Vallentin, and R. Sommer, "Here's my cert, so trust me, maybe? Understanding TLS errors on the web," in *Proc. of the 22nd International Conference on World Wide Web*, 2013.

[8] M. Acer, E. Stark, A. P. Felt, S. Fahl, R. Bhargava, B. Dev, M. Braithwaite, R. Sleevi, and P. Tabriz, "Where the wild warnings are: Root causes of chrome certificate errors," 2017.

[9] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman, "Towards a complete view of the certificate ecosystem," in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC '16. New York, NY, USA: ACM, 2016.

[10] M. O'Neill, S. Ruoti, K. Seamons, and D. Zappala, "Tls proxies: Friend or foe?" in *Proceedings of the 2016 Internet Measurement Conference*. ACM, 2016, pp. 551–557.

[11] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "RFC 5280: Internet X.509 PKI Certificate and Certificate Revocation List (CRL) Profile," Internet Engineering Task Force, Tech. Rep., 2008.

[12] ACME Working Group. (2017) Automatic certificate management environment (acme). [Online]. Available: https://tools.ietf.org/html/draft-ietf-acme-acme-09

[13] Comscore, "Comscore Privacy Policy," December 2017. [Online]. Available: https://www.comscore.com/About-comScore/Privacy-Policy

[14] "OpenSSL verify." [Online]. Available: https://www.openssl.org/docs/man1.0.2/apps/verify.html

[15] "cryptography 2.1.1." [Online]. Available: https://pypi.python.org/pypi/cryptography

[16] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar, "TLS in the wild: an Internet-wide analysis of TLS-based protocols for electronic communication," in *Proceedings of the Network and Distributed System Security Symposium*, 2016.

[17] (2013, August). [Online]. Available: https://www.telerik.com/blogs/faq—certificates-in-fiddler

[18] CloudFlare, "What We Just Did to Make SSL Even Faster," December 2012. [Online]. Available: https://blog.cloudflare.com/what-we-just-did-to-make-ssl-even-faster/

[19] R. Holz, L. Braun, N. Kammenhuber, and G. Carle, "The SSL Landscape: A Thorough Analysis of the X.509 PKI Using Active and Passive Measurements," in *Proceedings of the ACM Internet Measurement Conference*, 2011.

[20] E. Barker and Q. Dang, "Recommendation for Key Management, Part 3: Application-Specific Key Management Guidance," National Institute of Standards and Technology, Tech. Rep., 2015.

[21] (2005, Feb). [Online]. Available: https://www.schneier.com/blog/archives/2005/02/sha1_broken.html

[22] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full SHA-1," in *Cryptology ePrint Archive, Report 2017/190*, 2017.

[23] (2012, 10). [Online]. Available: https://www.schneier.com/blog/archives/2012/10/when_will_we_se.html

[24] (2014, September). [Online]. Available: https://blog.qualys.com/ssllabs/2014/09/09/sha1-deprecation-what-you-need-to-know

[25] T. Kleinjung, K. Aoki, J. Franke, A. Lenstra, E. Thomé, J. Bos, P. Gaudry, A. Kruppa, P. Montgomery, D. A. Osvik *et al.*, "Factorization of a 768-bit rsa modulus," in *CRYPTO 2010*, vol. 6223. Springer, 2010.

[26] "NIST Policy on Hash Functions." [Online]. Available: https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions

[27] "Gradually Sunsetting SHA-1." [Online]. Available: https://security.googleblog.com/2014/09/gradually-sunsetting-sha-1.html

[28] "Microsoft SHA-1 Deprecation Roadmap." [Online]. Available: https://blogs.windows.com/msedgedev/2016/04/29/sha1-deprecation-roadmap/

[29] "The End of SHA-1 on the Public Web." [Online]. Available: https://blog.mozilla.org/security/2017/02/23/the-end-of-sha1-on-the-public-web/

[30] (2013, June). [Online]. Available: https://bugzilla.mozilla.org/show_bug.cgi?id=881553

[31] (2014, September). [Online]. Available: https://blog.mozilla.org/security/2014/09/08/phasing-out-certificates-with-1024-bit-rsa-keys/

[32] [Online]. Available: https://www.globalsign.com/en/ssl-information-center/1024-bit-public-and-private-keys/1024-bit-public-and-private-keysfaq/

[33] [Online]. Available: https://www.symantec.com/page.jsp%3Fid%3D1024-bit-migration-faq

[34] Elaine Barker and Allen Roginsky, "Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths," Computer Security Division Info. Tech. Laboratory, Tech. Rep., 2011.

[35] Z. Durumeric, Z. Ma, D. Springall, R. Barnes, N. Sullivan, E. Bursztein, M. Bailey, J. A. Halderman, and V. Paxson, "The Security Impact of HTTPS Interception," in *Proceedings of the Network and Distributed Systems Symposium*, 2017.

[36] [Online]. Available: https://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=ca-certificates;dist=unstable

[37] [Online]. Available: https://fedoraproject.org/wiki/CA-Certificates

[38] [Online]. Available: https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/

[39] [Online]. Available: https://www.chromium.org/Home/chromium-security/root-ca-policy

[40] [Online]. Available: http://aka.ms/RootCert

[41] [Online]. Available: https://bugzilla.mozilla.org/buglist.cgi?component=CA%20Certificate%20Root%20Program&product=NSS&bug_status=__open__

[42] [Online]. Available: https://social.technet.microsoft.com/wiki/contents/articles/35984.microsoft-trusted-root-certificate-program-participants-as-of-october-14-2016.aspx

[43] [Online]. Available: https://bugs.openjdk.java.net/browse/JDK-8141540

[44] C. Evans, C. Palmer, and R. Sleevi, "Public key pinning extension for http," Google, Tech. Rep., 2015.

[45] (2018, January). [Online]. Available: https://chromium.googlesource.com/chromium/src/+/master/docs/security/faq.md#How-does-key-pinning-interact-with-local-proxies-and-filters

[46] "SSL Strip." [Online]. Available: https://moxie.org/software/sslstrip/

[47] "mitmproxy." [Online]. Available: https://mitmproxy.org

[48] "Charles Proxy." [Online]. Available: https://www.charlesproxy.com/documentation/proxying/ssl-proxying/

[49] P. Kotzias, A. Razaghpanah, J. Amann, K. G. Paterson, N. Vallina-Rodriguez, and J. Caballero, "Coming of Age: A Longitudinal Study of TLS Deployment," in *Proceedings of the Internet Measurement Conference 2018*. ACM, 2018, pp. 415–428.

[50] B. Amann, M. Vallentin, S. Hall, and R. Sommer, "Extracting certificates from live traffic: A near real-time SSL notary service," 2012.

[51] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by internet-wide scanning," in *Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security*. ACM, 2015, pp. 542–553.

[52] M. Acer, E. Stark, A. P. Felt, S. Fahl, R. Bhargava, B. Dev, M. Braithwaite, R. Sleevi, and P. Tabriz, "Where the Wild Warnings Are: Root Causes of Chrome Certificate Errors," 2017.

[53] O. Gasser, B. Hof, M. Helm, M. Korczynski, R. Holz, and G. Carle, "In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements," in *Int'l Conf. on Passive and Active Network Measurement*. Springer, 2018, pp. 173–185.

[54] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices." in *USENIX Security Symposium*, 2012.