Web PKI: Closing the Gap between Guidelines and Practices

Antoine Delignat-Lavaud[‡], Martín Abadi[†], Andrew Birrell[†], Ilya Mironov[†], Ted Wobber[†], and Yinglian Xie[†]
[†]Microsoft Research
{abadi,birrell,mironov,wobber}@microsoft.com
yinglian.xie@gmail.com
[‡]INRIA Paris-Rocquencourt*
antoine.delignat-lavaud@inria.fr

***Abstract*—A string of recent attacks against the global public key infrastructure (PKI) has brought to light weaknesses in the certification authority (CA) system. In response, the CA/Browser Forum, a consortium of certification authorities and browser vendors, published in 2011 a set of requirements applicable to all certificates intended for use on the Web and issued after July 1st, 2012, following the successful adoption of the extended validation guidelines in 2007. We evaluate the actual level of adherence to the CA/Browser Forum guidelines over time, as well as the impact of each violation, by inspecting a large collection of certificates gathered from Web crawls. We further refine our analysis by automatically deriving profile templates that characterize the makeup of certificates per issuer. By integrating these templates with violation statistics, we are able to depict the practices of certification authorities worldwide, and thus to monitor the PKI and proactively detect major violations. Our method also provides new means of assessing the trustworthiness of SSL certificates used on the Web.**

I. Introduction

For better or for worse, today's Internet is heavily reliant on its public key infrastructure to bootstrap secure communications. The current PKI, being the result of an extended standardization process, bears the marks of compromise: too few constraints on what certificates can express and too many parties wielding too much authority. These weaknesses are largely non-technical, since the X.509 certificate standard supports multiple mechanisms to constrain authority, for example on the namespace available to a given issuer. That such mechanisms are not generally used is due primarily to practical and business considerations. It therefore should not have been a surprise when such well-publicized exploits as the Flame malware [1] and the more recent misuse of a TÜRKTRUST certificate [2] targeted the PKI directly. In practice, it remains largely true that the security of the whole system is only as strong as the weakest certification authority (CA).

*This author was an intern at Microsoft Research during completion of this work.

Permission to freely reproduce all or part of this paper for noncommercial purposes is granted provided that copies bear this notice and the full citation on the first page. Reproduction for commercial purposes is strictly prohibited without the prior written consent of the Internet Society, the first-named author (for reproduction of an entire paper only), and the author's employer if the paper was prepared within the scope of employment.
NDSS '14, 23-26 February 2014, San Diego, CA, USA
Copyright 2014 Internet Society, ISBN 1-891562-35-5
http://dx.doi.org/doi-info-to-be-provided-later

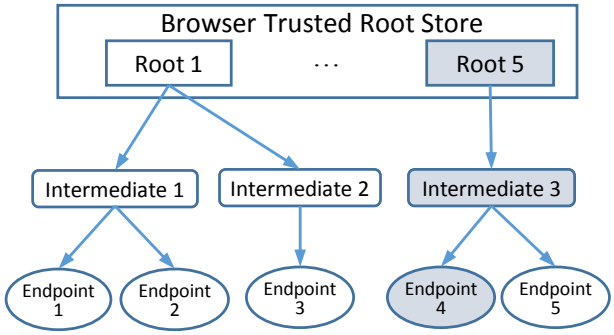

Fig. 1. Web PKI example, depicting trust chain from Root 5 to Endpoint 4.

Figure 1 depicts the trust relationships at play during Web browser certificate validation. Recall that browsers maintain a collection of trusted root certificates. This list is initialized and (usually) updated by the browser vendors. During TLS connection establishment, the target website offers an endpoint certificate referencing its domain name, as well as one or more intermediate certificates intended to allow the browser to construct a trust chain from one of its roots to the endpoint. In this context, certificate issuers are largely commercial entities, governments, or other large organizations; while Web browsers are the relying parties responsible for evaluating certificate trustworthiness. The details of which certificates should be trusted and for what purposes are considerably more intricate than this short description suggests [3], [4]. However, it remains largely true that any CA can issue certificates for anyone. This state of affairs amplifies the severity of problems that may arise.

Various attempts to augment or improve the Web PKI have followed. Google's certificate pinning and certificate transparency programs [5], [6], Convergence [7], and Perspectives [8] all introduce new mechanisms or services to better establish the trustworthiness of certificates. DANE [9] proposes to complement (or even replace) the trust anchor of the PKI with that of DNSSEC [10]. Needless to say, adopting any of these solutions requires substantial change to the relying parties responsible for certificate checking.

Perhaps the most fundamental changes to the PKI have

been undertaken by the certificate issuers and browser vendors themselves in the guise of the CA/Browser Forum. This forum has offered new, stricter guidelines on the issuance of certificates and the auditing process for CAs, building on existing mechanisms rather than replacing them. To a large extent, the success of this effort depends on compliance, since there isn't any client-side enforcement component specified as of yet, even though some browsers have recently taken steps in this direction.

Over two years have passed since the initial guidelines were adopted. Are they gaining acceptance? This paper offers an in-depth analysis of a large-scale collection of certificates as it evolves over time. In short, the answer to the question above is 'yes', there has been a significant degree of adoption. However, compliance is far from uniform and many violations persist. For example, there has been an order of magnitude improvement in the percentage of endpoint certificates that are furnished with identifiable policy statements by their issuers, but virtually no improvement in the number of certificates valid for local intranet names.

To understand the situation more precisely, we need to figure out how violations correlate with certificate issuers. Unfortunately, certificate issuance policies are far from consistent over time, even for a given CA. Thus, we extend our analysis to automatically derive per-issuer templates that characterize groups of certificates issued under a common policy, allowing us to measure compliance violations on a per-template basis. Grouping certificates into clusters enables us to see patterns and to identify specific templates in what is otherwise a big pile of fairly amorphous data.

The correlation between templates and violations not only allows us to better evaluate CAs with respect to compliance, but it also offers a new mechanism for determining whether a certificate seen for the first time matches an expected template with known compliance characteristics. So, for instance, if a new certificate appears for a given CA that has similar features to an existing cluster from this CA, and all other certificates in this cluster have a low level of compliance violations, then one might conclude that the new certificate is likely trustworthy. Conversely, a certificate that completely stands out, or matches a cluster of poor compliance behavior, might arouse suspicion. Hence, the clustering may also serve as a basis for policy enforcement. Furthermore, our experience is that a suitable visualization tool that factors in compliance violations is critical to understanding the current state of the Web PKI.

There have been a number of previous measurements of the deployment of TLS on the Web, both in terms of certificate quality and supported TLS cipher suites and extensions [11]–[16]. In particular, these studies have identified specific problematic certificates [17], [18] and quantified interesting patterns [12]–[17]. In this paper we aim to go beyond those previous studies by providing a detailed analysis of compliance with specific guidelines, and also in the development of particular analysis and visualization methods. We focus on only publicly trusted certificates and evaluate them under criteria adopted by all major root programs, thereby eliminating self-signed certificates from our dataset while ensuring that all uncovered violations are significant. When relevant, we compare the very recent results of Durumeric *et al.* [16] with our own, as they cover a similar time period, although the studies were carried out independently.

In summary, the contributions of this paper are:

- a principled, large-scale analysis of compliance with the CA/Browser Forum's guidelines over time;
- a new mechanism to automatically extract and validate templates that characterize certificate issuance policies;
- a compliance analysis and visualization tool for the inferred templates;
- the discovery, driven by policy violations reported by our tool, of exploitable vulnerabilities in some CA templates and certificate validation libraries.

The remainder of this paper is organized as follows: we first summarize the CA/Browser Forum guidelines applicable to this work in Section II. Section III describes how we collected the certificates for the study. Section IV presents our global compliance statistics and discusses the trends and impact of important violations. We detail our new template-oriented clustering method in Section V and discuss its results in Section VI. Finally, Section VII concludes and summarizes our findings.

II. Guidelines and Requirements

While the Web PKI presents users with a binary trust decision on the trustworthiness of certificates, the various issuance authorization processes entail significantly different levels of trust. A first step to address this issue was taken in 2010 with the adoption of the Extended Validation (EV) guidelines [19] and the implementation into Web browsers of a clear visual indication of the subject's verified identity for such high-trust certificates.

However, outside of EV certificates, there has historically existed considerable freedom in the way CAs managed and issued certificates. In the extreme case, some CAs made a business practice of selling intermediate authority certificates specifically intended for wiretapping encrypted connections [20]. An entity holding such a certificate, working with a transparent proxy, can sign certificates for arbitrary Internet entities that will be accepted by clients of the proxy, thereby subverting the normal function of the CA hierarchy.

Thus, it became apparent that stronger guidelines were required to maintain the sustainability of the growing PKI. In response, the CA/Browser Forum adopted the "Baseline Requirements for the Issuance and Management of Policy-Trusted Certificates" [21], a common standard applicable to all certification authorities that are trusted by major browsers. These guidelines make considerable strides toward regularizing certification practices. With respect to issued certificates, they provide significantly tighter constraints on cryptographic strength, certificate usage, revocation information, and signature authority among other things.

Yet, the baseline requirements remain a compromise between security and the continuation of existing business practices. For instance, while certificates issued to local names or IP addresses offer no authentication because they can be

moved from one local network to another, they are still allowed until October 2016. Similarly, there is as of yet no effective control over which names an intermediate CA can certify, thus increasing the potential for man-in-the-middle attacks whether well-intentioned or otherwise.

Today, publicly trusted CAs must comply with the following standards:

1) RFC 5280 [22], which describes the X.509 format as well as specific requirements about some certificate fields and extensions;
2) the rules of the Root Program where their root certificates will be installed: both the Microsoft [23] and Mozilla [24] root programs mandate yearly auditing by a third-party agency;
3) one of the following auditing standards: WebTrust for CAs (and optionally, WebTrust for EV Readiness), ETSI TS101 456 or TS102 042 or ISO 21188:2006.

The WebTrust and ETSI audit criteria both cover the baseline requirements starting from version 1.1 ([25], effective January 2013) for WebTrust and version 2.3.1 ([26], effective November 2012) for ETSI. We do not consider ISO 21188:2006 because none of the current authorities in the Mozilla Root program appears to be audited under that standard.

Since inclusion in root programs depend on the above audit criteria, we expect that all certificates issued after July 1st, 2012 (the effective date set by the CA/Browser Forum) should follow the baseline requirements, as well as the EV guidelines for extended validation certificates. There have been several revisions of the baseline requirements; for our evaluation, we chose to always consider the least restrictive condition found in all published versions.

The baseline requirements cover a broad range of topics: warranties, liability, the application and verification process, the safekeeping and protection of records, delegation, etc. We focus on the requirements that can actually be verified by certificate inspection: subject identity and certificate contents ([21], Section 9), certificate extensions ([21], Appendix B), and cryptographic algorithm and key requirements ([21], Appendix A).

A. Identity Verification and Contents

While there is no visual browser clue to distinguish low-trust and high-trust non-EV certificates, the CA/Browser profile requirements aim to allow clear identification of the issuer, subject, and issuance process of any certificate that the user may choose to inspect manually. Hence, there are distinctions on what information should appear in the issuer and subject of certificates based on the authorization method, as listed in Table I.

Certificates issued without any verification of the subject's identity, based on control or ownership of domains and IP addresses listed in the Subject Alternative Name extension, may not include an organization nor any location field. Such certificates are often referred to as *domain control validated*, or simply *domain validated*.

Certificates for which the CA has conducted verification of the organization or individual identity may include an organization name, as well as any location information that was also verified. Such certificates are colloquially known as *organization validated*.

TABLE I. X.500 NAME REQUIREMENTS.

X.500 Issuer Fields	
Organization	Required; a name or trademark that identifies the issuing CA
Country	Required; code of country where the CA business is located
Common Name	Optional; if present, should accurately identify the issuing CA

X.500 Subject Fields	
Common Name	Deprecated, must contain a single IP or FQDN if present Subject Alternative Name extension must list applicable names
Organization	Optional, may only appear if verified by the CA Required for extended validation certificates
Location	Covers the Street Address, Locality, State and Postal Code fields Must appear if an Organization name is listed, mustn't otherwise Location must be verified by the CA if present
Country	Required if an organization is listed, must match its location If no organization is listed, may appear based on - the top-level domain of one of the applicable domain name; - IP geolocation of either an applicable IP or the applicant
Registration	Covers Business Category, Incorporation Locality/State/Country Required for extended validation, may not appear otherwise Registration number must also appear in Serial Number field

Finally, if the CA has conducted the extensive identity and incorporation verification process described in the EV guidelines [19], among other technical requirements, it may issue an *extended validation* certificate which will cause browsers to display the subject's verified identity prominently.

Besides the fields listed in Table I, the subject may include a valid sequence of domain components, and arbitrary unverified values in the Organizational Unit field if they cannot be confused for a name, trademark, or address. Other fields may be included as long as their values are verified in the issuance process.

Finally, another concern with the certificate system stems from changes in subject identity or control over listed names and addresses after the certificate issuance. The only response to this issue is to restrict the maximum validity period of endpoint certificates to 5 years, a limit that will drop to 39 months on April 2015. EV certificates may not be valid for more than 27 months.

B. Cryptographic Requirements

The CA/Browser Forum allows RSA, DSA, and EC keys in certificates. RSA keys should be at least 2048 bits long, with three exceptions for 1024-bit keys: endpoint certificates that expire before 2014; intermediate CA certificates issued before 2011 and expiring before 2014; and root certificates issued before 2011 that directly sign endpoint certificates. CAs should also ensure that the modulus has no factors smaller than 752, is not a power of a prime, and is not known to be vulnerable (e.g., due to the Debian OpenSSL bug [27]), and that the exponent is an odd number in the range $[2^{16}+1, 2^{256}-1]$.

All DSA keys should be at least 2048 bits long with 224- or 256-bits divisor. Furthermore, CAs must check the order of the generator and the representation of the public key of all certificates they sign.

Supported elliptic curves are NIST P-256, P-384, and P-521. CAs should use the partial or full ECC Public Key Validation Routine described in NIST SP 800-56A [28] to check the validity of public key from applicants.

TABLE II. EXTENSIONS OF ENDPOINT CERTIFICATES.

Extension	Requirements
Certificate Policies	Must appear, should not be critical Must include the OID of the issuer's policy May include link to online CPS on issuer website
CRL Distribution Points	Must appear, should not be critical Must include HTTP URL of issuer's CRL file
Authority Information Access	Must appear, must not be critical Must contain HTTP URL of issuer's OCSP service Should contain HTTP URL of issuer's certificate
Basic Constraints	May appear, must be critical if present CA flag must be set to false
Key Usage	May appear, should be critical Must not include "Certificate/CRL Signature"
Extended Key Usage	Must appear, may be critical Must include "Client/Server Authentication" May include "Email Protection" Should not include any other value
Subject Alternative Name	Must appear Should not be critical, unless subject is empty Must include subject's Common Name, if present Must only contain DNS names and IP addresses Should not contain local names or IP addresses

Supported digest algorithms are SHA-1, SHA-256, SHA-384, and SHA-512, with the exception of root certificates issued prior to 2011, which may be self-signed using MD5. There is no requirement about the signature algorithm to use with RSA and EC keys but in most cases PKCS#1 v1.5 and ECDSA are used respectively.

Finally, serial numbers must be non-sequential and contain at least 20 bits of entropy.

C. Certificate Extensions

Depending on the intended use of the certificate (root, intermediate CA, or endpoint), the baseline requirements mandate different constraints on the extensions that they should include, as well as their semantics. Together, those checks aim to satisfy the following goals:

- enforce the ability to assess the precise issuance policy of every certificate in a trusted chain;
- facilitate the reconstruction of chains that are invalid or missing some intermediate CA certificates;
- ensure the ability to efficiently check the revocation status of every certificate in a trusted chain;
- prevent any attack resulting from variations in implementation or supported features of different certificate validation software.

Not all implementations of certificate chain validation fully support all standard extensions. For instance, the *name constraints* extension, which can restrict the namespace of domains that a CA can sign certificates for, is not yet supported in all browsers. To prevent such security-critical restrictions from being ignored because they are not implemented, a *critical* flag can mark extensions for which lack of support must cause rejection of the certificate chain.

The precise requirements for each certificate category are listed in Tables II, III, and IV.

In addition, certificates should not include any extension, key usage or extended key usage flag that is not listed in the above tables without a specific reason. For this last requirement, we can only evaluate how often additional extensions or key usages are added by CAs, regardless of the purpose of inclusion.

TABLE III. EXTENSIONS OF INTERMEDIATE CA CERTIFICATES.

Extension	Requirements
Certificate Policies	Must appear, should not be critical Must include the OID of the CA's issuance policy May include link to online CPS on issuer website
CRL Distribution Points	Must appear, should not be critical Must include HTTP URL of this CA's CRL file
Authority Information Access	Must appear, must not be critical Must contain HTTP URL of issuer's OCSP service Should contain HTTP URL of issuer's certificate
Basic Constraints	Must appear, must be critical CA flag must be set to true Path Length constraint may be set
Key Usage	Must appear, must be critical Must include "Certificate" and "CRL Signature" May include "Digital Signature" for OCSP signing
Name Constraints	May appear, should be critical if present

TABLE IV. EXTENSIONS OF ROOT CA CERTIFICATES.

Extension	Requirements
Basic Constraints	Must appear, must be critical CA flag must be set to true Path Length constraint should not be set
Key Usage	Must appear, must be critical Must include "Certificate" and "CRL Signature" May include "Digital Signature" for OCSP signing
Extended Key Usage	Must not appear

III. MEASURING THE CERTIFICATE ECOSYSTEM

In this section, we present the data collection and preprocessing steps for our study, and compare it with previous measurements. Given the distributed and evolving nature of the Web PKI, collection efforts limited to a single time period or locale are unlikely to yield a complete picture necessary for implementing needed changes. Instead, our goal is to develop a scalable infrastructure for investigating practices of individual CAs, not in the sense of a business entity, but as a single issuer of endpoint certificates, to better reflect delegation by means of intermediate authority certificates.

A. Data Collection

The most common ways of collecting certificates are exploration of the IPv4 address space (as conducted for the 2010 Electronic Frontier Foundation's SSL Observatory [18], or with a fast scanner such as ZMap [29]), crawling of a list of known websites (such as Alexa Top 1 Million [30]), and gathering certificates used by a large set of users, either on their system or by inspecting live traffic on the network (e.g., the ICSI certificate notary [31]).

We use the data collection methodology from Abadi *et al.* [17], which is based on the combined crawl of the EFF's SSL Observatory IP addresses and the Alexa Top 1 Million websites. The total data set contains 8,349,808 unique certificates, but for our evaluation, we focus on the ones that are publicly trusted and issued in the two-year window before and after the effective date of the baseline requirements (July 1st, 2012), which amounts to 1,480,028 certificates. The last crawl for the data collection process occurred on July 31, 2013.

It is worth noting that the Alexa Top 1 Million list does not distinguish different subdomains of the same website: the only subdomain that we attempt to connect to is `www`, because it is almost universally used. We rely on the IP address crawl to capture other subdomains. Similarly, like previous measurements that rely on crawling of IPv4 addresses, we do not collect different certificates that are served on the same IP address based on the value of the *server name indication* (SNI) extension in the TLS handshake, except those that are used by websites in the Alexa list. For those, we can include the domain name in the SNI.

B. Path Reconstruction

During the data collection phase, we store all the certificates returned during the TLS handshake regardless of whether they form a valid or complete chain. Unlike most previous HTTPS measurements, we are interested only in the behavior of CAs but say nothing about how the issued certificates are being deployed in Web servers. Deployment statistics can be found in more general studies [12], [13].

After the certificates have been collected, we completely recreate the signature binary relation offline and store it along with path reconstruction information such as subject, issuer, and key identifiers (if present) in indexed SQL tables. As explained in Section II, the baseline requirements apply to chains rather than individual certificates. When evaluating a certificate, we consider all the valid paths to a root, and use several heuristics to select a path that uses the most recent version of each CA certificate. We observe that it is not uncommon to find different CA certificates that share the same key (and sometimes, the same subject name as well), with some versions showing better compliance, or signed with a stronger algorithm. Our reconstruction algorithms aims to ensure that we measure the most compliant chain that is possible to obtain from the different versions of root and intermediate CA certificates that have been distributed. Because the CA/B Forum guidelines surrounding certificate extensions are frequently violated, it is occasionally not obvious whether, for instance, a certificate was issued for CA or endpoint purposes. In order to decide which class of requirements should be enforced, we infer each collected certificate's type from its position in the reconstructed chain.

IV. Global Evaluation

In this section, we evaluate the compliance with guidelines and requirements from Section II of each collected certificate along with its reconstructed trust chain. We present the clustering method and results in the next two sections. We consider the two one-year periods before and after July 1st 2012, the effective date of the baseline requirements. We harvested 809,425 publicly trusted certificates issued during the first period signed by 744 distinct intermediates, and 670,603 trusted certificates signed by 668 intermediates after the date.

Overall, in the year before the effective date, just 0.39% of issued certificates strictly adhered to all the baseline and extended validation guidelines. In the following year, that number rose to 0.73%, all of which are extended validation certificates. We now detail each category of violations and discuss their impact.

A. Names Violations

Our first evaluation covers the applicable names of certificates. A notable trend between the two evaluation periods is the increased number of names each certificate is valid for, which rose from 1.96 to 2.2 on average. The share of certificates containing distinct second-level domain names (i.e., `a.x.com` and `b.y.net`, but not `a.x.com` and `b.x.net`) grew from 52% to 56%. We further discuss this observation in Section VI.

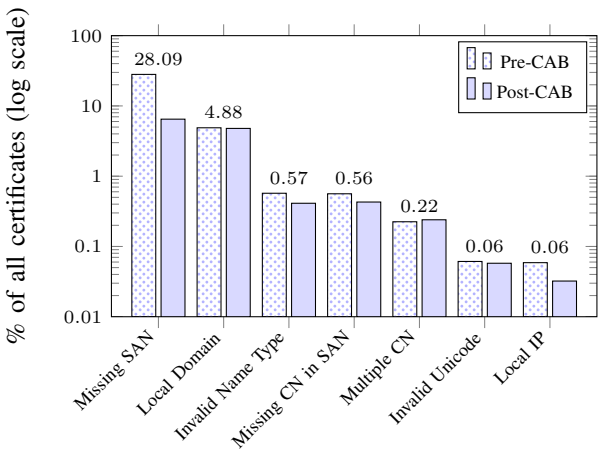

Fig. 2. Subject Name Violations

In terms of violations, we find that the proportion of certificates that lack the required subject alternative names (SAN) extension decreased sharply from 28.09% to only 6.48%, as shown in Figure 2. In parallel, the share of certificates that contain at list one wildcard name increased from 9.2% to 12.3%.

In Figure 2, we also observe that close to 5% of Web certificates are valid for local names and IP addresses. Thus, intranet certificates still seem to constitute a large market for CAs, despite the fact that such certificates do not offer any authentication, as we previously mentioned. In fact, mixing Internet and local names is not technically considered a violation of the baseline requirements until 2016.

As for the other violations, we noticed some unusual name types (most often email addresses) in 0.4% of certificates, and Unicode names that were rejected by our Internationalized Domain Name (IDN) decoding library in 387 instances. The baseline requirements recommend checking for IDN names that may be used for phishing [32] (which is difficult to detect because of graphical similarities between some Unicode characters and letters of the Latin alphabet), but without further details we were not able to perform additional checks for this requirement.

B. Issuance and Subject Identity Violations

We now examine requirements related to the issuance process and subject identification. The market share of each validation process stayed relatively stable, from 48% domain validated, 48% organization validated, and 4% extended validation certificates to 49.2%, 46.6%, and 4.2%, respectively.

In Figure 3, we observe significant improvements overall. Most notably, only 1.75% of recently issued certificates are

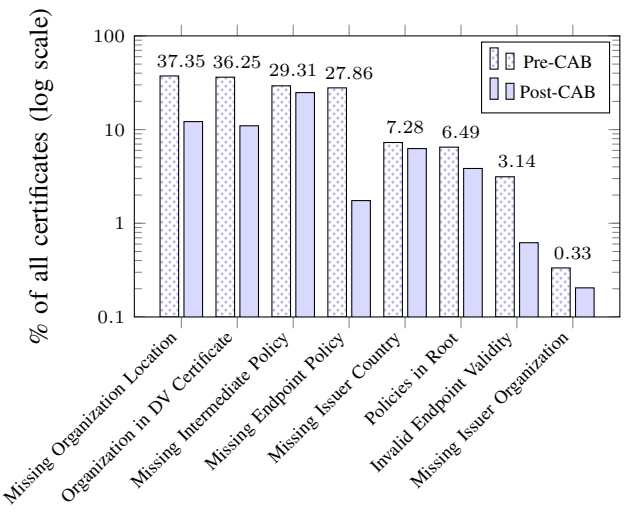

Fig. 3. Identification and Issuance Violations

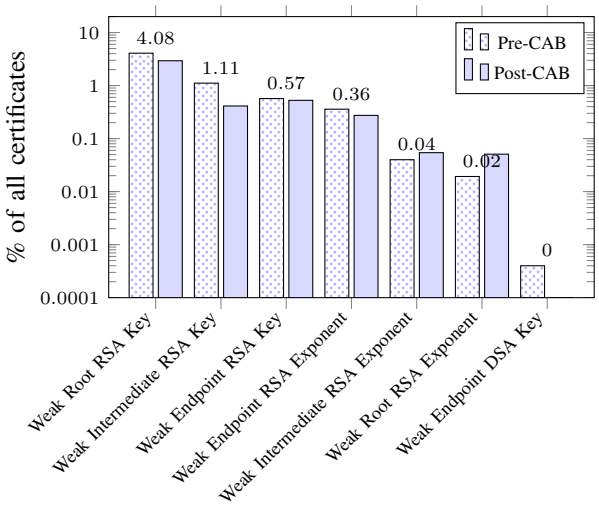

Fig. 5. Cryptographic Violations

still missing their issuance policy, compared to 27.8% just one year before. Second, the number of issued certificates valid for a duration longer than the CA/Browser Forum's limits also went down sharply. Overall, there is a clear positive trend towards better identification of the subject and issuer of a certificate. Confusion and phishing risks are also reduced by not including subject fields (such as an organization name) that are not verified as part of the issuance process. Unfortunately, these improvements do not directly benefit end users because of the lack of visual clues in browsers, except for extended validation certificates.

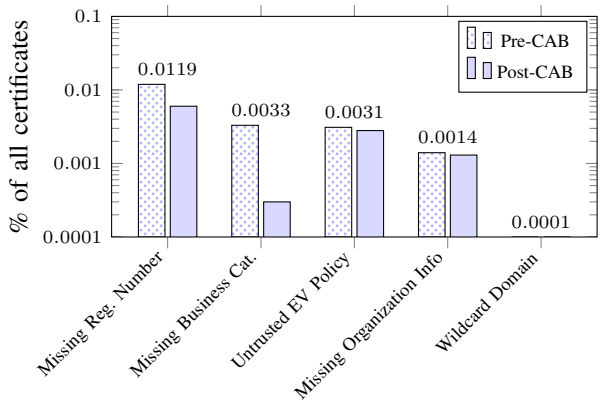

Fig. 4. EV Guidelines Violations

We observe only a very small fraction of violations of the extended validation guidelines, as shown in Figure 4, suggesting the concrete impact of standardized rules. In particular, all of the certificates that showed complete adherence with all applicable standards have been issued with extended validation.

C. Cryptographic Violations

In this section, we evaluate adherence to the cryptographic requirements described in Section II. Figure 5 shows the statistics of each violation.

Among the certificates we collected, all but three use RSA for their public key, with an average modulus size increasing from 1921 to 2017 bits between the two time periods. While there are some elliptic curves certificates in use on the Web, for instance by Google, they are typically presented during a TLS handshake only if the initial client message demonstrates EC support. Out of the three DSA certificates from 2011–2012, two use a 1024-bit modulus, while the third has 512 bits. They are now expired and DSA doesn't seem to be used on the Web anymore.

In terms of key lengths, perhaps surprisingly, we find that the proportion of signed certificates with 1024-bit keys actually went *up* from 4.3% (plus 117 intermediate CAs) to 5.2% (plus 2 intermediate CAs) between the two periods. For endpoint and intermediate CA certificates, 1024-bit keys are allowed by the CA/Browser Forum if they expire before 2014. Checking this requirement, the percentage of violations among endpoint certificates is in fact going down slightly from 0.57% to 0.53%. Investigating further, we found that the main providers of 1024-bit keys (Google, Akamai, and Servision) are issuing only short lifespan certificates and seem to be in the process of moving to 2048-bit keys.

We did not find any endpoint certificate issued after July 1st, 2011 that was signed with MD5. Adoption of the SHA-2 family of hash functions also increased from 0.2% to 0.6% between the two evaluation periods, and we found no vulnerable key caused by the Debian OpenSSL bug [27] in publicly trusted certificates issued since 2012.

Low RSA exponents constitute a potential risk when the relying party fails to implement a correct validation of signatures formatted according to the PKCS#1 v1.5 standard [33]. Although the level of compliance with this requirement is already very high (exceeding 99.5%), it has improved only marginally over the observation period.

D. Extension Constraints

We now move to the violations of constraints on the extensions that a certificate should include (Tables II, III,

and IV). Violations of this type are usually more security sensitive. In particular, because of the complexity and fragility of the requirements that govern the constraints in a certificate chain, not all popular libraries for certificate validation apply the necessary checks consistently and in full compliance with current standards. While major Web browsers generally behave adequately, a significant fraction of HTTPS requests are now performed by libraries as part of other programs. Such libraries are often much less scrutinized and they tend to rely on their underlying TLS libraries for certificate validation [4]. Conversely, TLS libraries may delegate too many checks to applications. For instance, all versions of GnuTLS ignore unsupported critical extensions. Even when such omissions are documented, it is not reasonable to expect application developers to validate anything besides the domain name correctly. Such misunderstandings between TLS libraries and applications are especially worrisome and deserve further investigation in future work.

Since the constraint requirements depend on the certificate type (root certificates, intermediate CA certificates, endpoint certificates), we discuss them separately below.

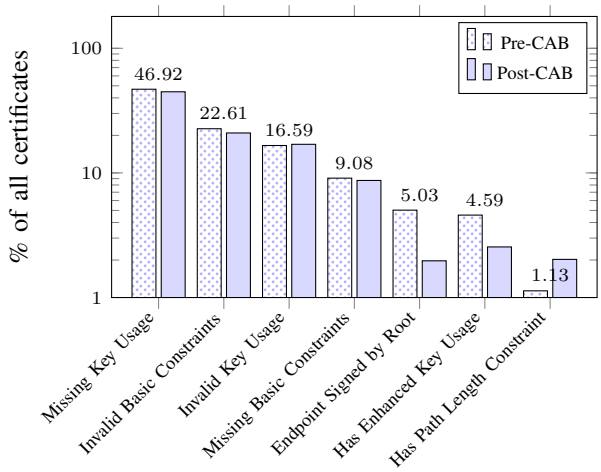

Fig. 6. Extension Violations in Root Certificates

1) Root Certificates: We first look at violations in root certificates, shown in Figure 6. Since a majority of the root certificates have been issued years before the baseline requirements went into effect, it is not surprising to find a large number of violations.

First, 29.6% (down from 31.6%) of chains either have invalid basic constraints in the root, or are missing basic constraints altogether. This extension is used to indicate whether the certificate has CA capabilities. If it does, it can further specify whether to restrict the maximum length of a valid chain rooted at this certificate, a feature known as *path length constraint*. The baseline requirements mandate this extension to be marked critical, with the effect of forcing any chain validation software to reject the certificate if it does not fully support the extension.

Including a path length limit in a root certificate is considered a violation by the CA/Browser Forum, which we found in 2% of chains (up from 1.1%). The rationale for this requirement is not given and, indeed, not clear: while most roots are expected to only ever sign intermediate CA certificates offline, limiting the path length to 0 is certainly a good idea for the six remaining roots that issue endpoint certificates.

Second, almost half (44.7%, down from 46.9%) of the root certificates do not include the *key usage extension*. This extension restricts how the certificate may be used to a subset of predetermined purposes, the most common being digital signature, non-repudiation, key encipherment, data encipherment, key agreement, certificate signing, and CRL signing. For HTTPS over TLS, digital signature and key encipherment flags are sufficient. If no key usage extension is present, the certificate is valid for all purposes.

Because key usages are limited to a fixed set of values, the *extended key usage* extension can enable additional purposes, indicated by an arbitrary number of custom Object Identifiers (OID). For instance, code-signing certificates attached to signed Java and Windows programs must include specific OIDs in addition to the digital signature key usage. About 2.5% (down from 4.6%) of chains violate the requirement not to include the extended key usage extension in a root certificate.

The justification for this requirement follows from the semantics of this extension, which are drastically different from key usage because they affect other certificates in the chain. First, for an extended key usage to apply to a certificate, it must appear in the metadata of the root certificate of its chain, as set by the root program manager. Hence, both Mozilla and Microsoft include with each root certificate a list of extended usages they are valid for, such as S/MIME, code signing, or document signing. Then, any certificate on a trusted chain that contains this extension restricts the set of possible extended usages of all its descendants to be a subset of the ones listed in its extended key usage extension *if the field is present*. A side effect of this enforcement algorithm is that the leaf of a chain where this extension never appears inherits all extended key usages from its root.

The large number of root certificates that have violations on basic constraints, path length constraints, and key usage extensions leads to a surprising observation: several of the trusted roots are actually not valid for CA purposes according to RFC 5820. This fact means that chain validation software must implement exceptions for accepting root certificates that are sometimes missing the basic constraints or key usage extensions altogether. There are means of correcting this situation, as it is in fact possible to "update" a root certificate while keeping the same key, a procedure used no less than three times on a major Verisign root certificate since 1999.

2) Intermediate CA Certificates: For intermediate CA certificates, the overall situation is more reassuring, as shown in Figure 7. All instances of basic constraints and invalid key usage are due to the extensions not being marked critical as required.

While these results may appear good, the main cause for the low number of violations is the lack of sufficiently strict requirements for intermediate certificates. For instance, it would make sense to require that every endpoint-issuing intermediate CA to have a path length constraint of 0. Fortunately, an increasing number of CAs are taking this precaution, up from 40% of intermediate CA certificates issued during the first

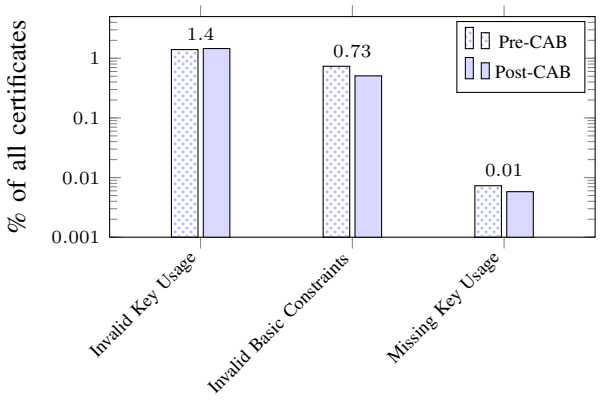

Fig. 7. Extension Violations in Intermediate Certificates

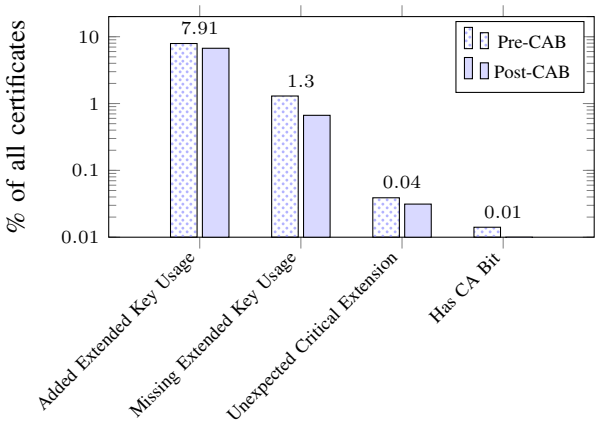

Fig. 8. Extension Violations in Endpoint Certificates

period to 80% For comparison, the value reported in [16, Section 5.6] is 43%.

Similarly, among the hundreds of intermediate CA certificates, many are issued to corporations (not only banks or Internet service companies, but also retailers, industry, entertainment...) that should not need to hold signature power over the entire Internet namespace. Only 11 active intermediate certificates include the *name constraints* extension to limit their scope, and they have signed a mere 44 certificates since July 2011. Durumeric *et al.* [16] report finding 7 such intermediate CA certificates active since March 2013.

While RFC 5820 requires that CA certificates have the key usage extension, the baseline requirements do not recommend adding extended key usage restrictions in intermediate CA certificates. Since public CAs mostly sign certificates for use on Web servers, there is no harm in adding an extended key usage restriction containing only the necessary "client authentication" and "server authentication" usages in an intermediate CA certificate, and it can prevent accidental usages being enabled on endpoint certificates that are missing the extended key usage extension.

Finally, RFC 5280 provides two mechanisms for specifying whether a certificate can be used as an intermediate: the basic constraints extension, which must be present and have the CA bit asserted, and the key usage extension that, if present, must include the `keyCertSign` bit. Current versions of OpenSSL (1.0.1e, as well as the development branch, 1.0.2) accept a certificate as a valid intermediate on the basis of the key usage extension alone if the basic constraints extension is missing [34]. Issuers that deviate from the baseline requirements by not including the basic constraints extension run the risk of accidentally creating a certificate endowed with the signing power when interpreted by OpenSSL if the certificate includes the key usage extension asserting the `keyCertSign` bit.

3) Endpoint Certificates: Moving on to endpoint certificates in Figure 8, we find that the most striking violation for endpoint certificates is the presence of the CA bit. Although only a small fraction (1.4%, all issued before July 2012) of endpoint certificates have this violation, the corresponding Web servers holding the private keys could use their certificates as a CA and sign arbitrary trusted certificates.

This violation is especially worrisome considering that, in January 2013, a certificate that had been incorrectly issued by the Turkish authority TÜRKTRUST was used to mount man-in-the-middle attacks against Google services [2]. Between 2010 and 2011, an intermediate authority on the Government of South Korea root issued at least 1580 endpoint certificates to Korean schools, universities and organizations with CA capability. Durumeric *et al.* [16] report finding 1395 of these certificates. 114 of them have been issued after July 1st, 2011 and two years later, 111 of them have not yet expired, and several dozen use easily factorable 512-bit public keys, as recently demonstrated [35].

In addition, some of these endpoint certificates with CA capabilities do not include the key usage extension, although it is not mandated by the baseline requirements. Fortunately, the intermediate issuer of these certificates had a path length constraint of 0 in its critical basic constraints extension, which should prevent any malicious use in compliant X.509 implementations. Yet, this safeguard is not required by the CA/Browser Forum, and we found evidence of incorrect chain validation implementations. Thus, the violation statistics support the need for stronger intermediate CA certificates constraints. Furthermore, we found that the GnuTLS library prior to version 3.0, still in common use today, ignore the path length extension in validating certificate chains. Thus, while limiting the path length to 0 for issuing intermediates is a good defense in depth, it may not prevent exploitation of rogue CA certificates in all clients.

We also found a non-negligible fraction of violations related to the extended key usage extension. For endpoint certificates, the use of additional extended key usages is not recommended by the baseline requirements, except in a few cases (e.g., for Server Gated Cryptography, an obsolete cryptographic enhancement standard used to bypass US export restrictions on strong cryptography in the 1990s). More importantly, we found 2064 Web certificates that were explicitly valid for code signing, and 3917 certificates that wrongly include the special "Any Key Usage" OID. However, it is not clear what software actually honors this value for extra purposes.

We also observed that around 1% of currently valid certificates are missing the extended key usage extension altogether. This violation is serious because, as explained previously, if the

extended key usage extension never appears in a trusted chain, the endpoint certificate inherits all extended key usages from the metadata of the root certificate of the chain, potentially making the certificate valid for S/MIME, code signing, document signing, network authentication, etc. Thus, it is essential for security to include this extension, and we advocate to also add it to intermediate CA certificates that only issue Web certificates as an extra safeguard, even though this is not permitted by the CA/Browser baseline requirements.

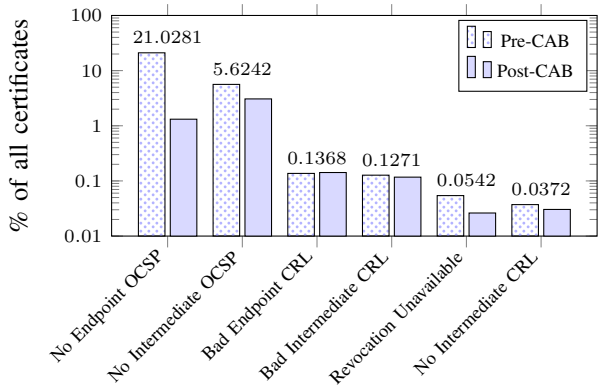

Fig. 9. Revocation Violations

4) Revocation Violations: We further examine violations of the extension requirements related to revocation (Tables II and III) in Figure 9. Revocation availability is an area that shows significant improvement. We observe a dramatic decrease of violations, in particular, much broader availability of the Online Certificate Status Protocol (OCSP), from 79% of certificates to 98.7%. OCSP has an important advantage over revocation lists: it forces CAs to record the serial numbers of certificates they have issued, and the OCSP server may only indicate that a given serial number is valid if it appears in the CA's records. Furthermore, the use of OCSP stapling [36] can improve latency caused by revocation checking. Overall, the total number of certificates for which we were not able to check the revocation status by any means went down from 439 to 176, from 13 different issuers (a value consistent with [16]).

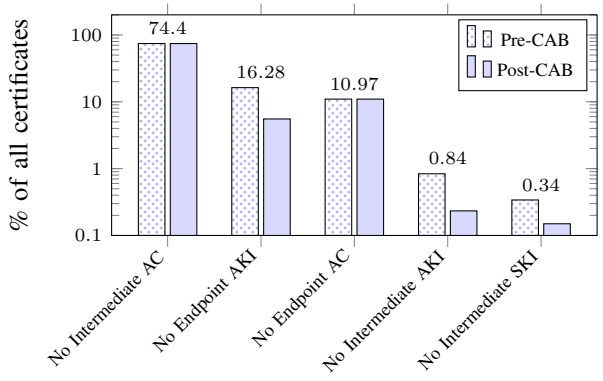

Fig. 10. Path Reconstruction Violations

5) Path Reconstruction Violations: We now examine the set of requirements meant to facilitate path reconstruction. We also observe quite a bit of improvements in this area comparing the two periods. In Figure 10, we show violations in three extensions that can help chain reconstruction: *subject key identifier* (SKI), *authority key identifier* (AKI), and *authority information access* (AIA). The AIA extension should contain two URIs: the issuer's OCSP responder and the *authority certificate* (AC) file in case it is missing from the presented TLS chain and not available on the system, in particular when updates to CA certificates cause subject and issuer name mismatches. SKI contains a unique identifier for the embedded public key (usually, it is the SHA-1 digest of the raw RSA modulus), and AKI should contain the same value as the issuer's SKI. These extensions can speed up chain reconstruction by storing an index of their values along with root and intermediate certificates.

V. Template-level Analysis

The individual chain violation statistics from Section IV present two major limitations:

1) They do not yield any insight on the individual practices of each certificate issuer. While we obtain statistics on the mass of all certificates, we also need an automatic method to provide a global picture of where the major vulnerabilities originate, and who is responsible for fixing them.
2) Their level of granularity does not enable a systemic evaluation of the CA infrastructure. It is common practice for a CA to delegate its signature power to a third-party organization by issuing it an intermediate CA certificate. Such delegated authorities are supposed to follow the same constraints as root authorities. We found at least 634 intermediate certificates that were used to sign at least one certificate since July 1st, 2012. For instance, the GTE CyberTrust Global Root, operated by Verizon, signs no less than 40 intermediates, all but 3 of which are managed by other organizations. A challenging issue for our analysis is to measure the difference in compliance of third-party delegated authorities compared to the root operators.

In this section, we present a new analysis method based on the simple idea that many of the baseline requirements apply to *CA profiles* rather than to individual certificates.

A. Template Clustering

Virtually all CAs use issuance profiles to sign endpoint certificates, which include information such as the format and entropy source of serial numbers, the fields in the X.500 subject name, the allowed validity periods, the signature algorithm, and the set of X.509 extensions that will appear in certificates issued with that profile. This information normally appears in the CA's Certificate Policy Statement (CPS). As a general rule, different profiles are used depending on the certificate purpose and validation method. For instance, all endpoint certificates must include an HTTP URI pointing to its issuer's Certificate Revocation List (CRL) in the CRL Distribution Points extension; if a profile includes this extension, this requirement will be met by all certificates issued with this template.

Since the CPS format is not usually machine readable, we aim to reconstruct profile information by running a clustering

algorithm over certificates represented as vectors of features. We pursue two separate goals in applying clustering to the set of certificates. First, by grouping together certificates issued by similar processes, we reduce the complexity of the certificate universe and allow manual inspection of characteristic representatives, thereby addressing the first challenge. Second, we can compare the guideline violations found in each cluster, allowing us to measure differences in compliance between CAs and their third-party delegated intermediate authorities, as well as among each other, thus addressing the second challenge.

B. The Clustering Algorithm

In order to apply a clustering algorithm, we choose a distance measure over vectors of features extracted from certificates. The distance between two certificates is defined as a weighted sum of distances between corresponding features. The relevant features can be numerical (e.g., the certificate's validity period), categorical (e.g., the signature algorithm), and attribute sets (e.g., extensions). For each class of features we define a distance function: the L_1 metric for numerical features, the discrete metric for categorical features (i.e., $d(x, y) = 1$ iff $x = y$, 0 otherwise), and the Jaccard distance for sets ($J(A, B) = 1 - \|A \cap B\|/\|A \cup B\|$).

The weights are assigned to the features in accordance to their relative importance in evaluating certificate similarity: high-weight features should all have the same exact value within a given cluster (for instance, the CA bit), medium-weight features should have few variations, while low-weight features can have a broad range of values but are useful in evaluating the "tightness" of each cluster. The features for each weight class are given in Table V.

TABLE V. CLUSTERING FEATURES.

High Weight	Medium Weight	Low Weight
Parent CA	Subject name fields	Key size
Signature and key algorithms	CRL distribution points	Issuance date
Set of X.509 extensions	Extended key usage	Validity period
Policy identifiers		Serial number length
Authority information access		
Key usage, basic constraints		

We evaluate the quality and robustness of our selection of distance measures and feature weights by comparing it with other methods. Specifically, we tried: using the L_2^2 measure for numerical features; setting the weights uniformly; setting weights to be inversely proportional to the standard deviation of the corresponding feature (thus normalizing the relative contribution of each feature to the aggregate distance). For each choice of distance measures and feature weights, we compute the distribution of rule violations in the resulting clusters, and select the setting that produces the most bi-modal distribution (while keeping the number of clusters constant). This procedure seeks to improve the predictive value of grouping by maximizing the number of clusters where certificates either all share a particular violation or none do. Yet, we found that none of our attempts to change the weights and distances improves the violation distribution significantly compared to our *ad hoc* weights in Table V and the L_1 metric.

The clustering procedure applies the k-medoid algorithm seeded with the k-means++ initialization step. The important guarantee of the k-medoid algorithm is that cluster centers (exemplars) are always members of the input dataset, which greatly facilitates subsequent analysis.

C. Cluster Evaluation

The clustering step aggregates CA profiles based on their similarity. After clustering, we perform the following evaluations for any cluster that generates template violations:

First, we perform the checks from Section II on the center of each cluster, and record any violations.

Second, for each certificate in the reported cluster with violations, we check a set of baseline requirements that apply to individual certificates rather than to templates, for example, the key size and validity period, the conformance of subject fields and subject alternative names, or the revocation status. This step collects statistics about such violations within each cluster. It provides useful feedback both about the quality of the cluster (e.g., if a large proportion of certificates are revoked, something may be wrong with the template) and about the relevance of the clustering (since we expect that for a given template, a given certificate-specific violation is either very frequent or very rare).

Finally, for each cluster with template-specific violations, we additionally examine the validity of certificate domains and the corresponding IP geolocations. In particular, we perform the following set of checks:

1) look up WHOIS information to compare the domain owner information with the subject fields, and the creation and expiration date of the domain and certificate;
2) resolve each listed domain name with DNS to ensure they are active and obtain their IP address;
3) check whether the IP address geolocation matches the country listed in the certificate;
4) check the revocation status of the certificate.

These additional examinations require network queries and cannot scale to millions of certificates. Thus, for each cluster, we randomly sample at most 1000 certificates on which to perform extended evaluation. Furthermore, the collected information is not always reliable. Still, we record the percentage of each violation along with the template-level violations for manual examination, as an additional source of feedback for our evaluation.

VI. CLUSTERING RESULTS AND DISCUSSIONS

We now present the results of running the analysis method described in Section V on certificates issued after the effective date of the baseline requirements. In §VI-A, we describe a visualization tool we designed to depict the results of our clustering analysis described above. In §VI-B, we investigate the relationship between the size of a CA and its level of compliance. In §VI-C, we investigate violations that can be detected with DNS queries for each domain name listed in a given certificate. In §VI-D, we highlight a new class of certificates used by content delivery networks. Finally, we show how to estimate the entropy of serial numbers in a given cluster in §VI-E.

A. Visualization of Results

Our clustering algorithm produced 571 clusters containing more than 5 certificates for the one-year period starting from July 1st, 2012. In order to present results in an intuitive, comprehensible format, we built a visualization tool that allows interactive exploration of the graph of certification authorities and template clusters. It implements the following features:

- search by CA name;
- detailed inspection of clusters, including the complete reconstructed template, the number of certificates, the average issuance date for the cluster, references to the center and a few sample certificates from the cluster, the set of violations for the center and the distribution of individual certificate violations from 1000 random certificates from the cluster;
- filtering of clusters based on the presence of some violation;
- assignment of custom scores to each template based on individual violation, and coloring of clusters based on the total score of their template and individual violations.

Figure 11 demonstrates the interface of our visualization tool. Each node of the depicted graph corresponds to either a CA certificate, if the node has an outline circle, or a cluster otherwise. This graph consists of a forest of disconnected trees where roots, leaves, and connecting nodes respectively correspond to root certificates, clusters, and intermediate CA certificates, while edges denote the signature relation between them. The area of a node is proportional to the sum of the sizes of clusters that are reachable from this node in the graph. Labels denote either the name of root and intermediate CA nodes, or the normalized average distance to the center of cluster nodes. The left panel offers an interface for certificate searching and cluster inspection, currently showing the search results for "DFN-Verein PCA Global - G01", a German CA marked by an arrow on the right side of the figure. While this CA is used in a small number of chains (as indicated by the size of its node), it signs a very large number of intermediate CA certificates (the connected nodes laid out in concentric circles). We further discuss delegation questions in §VI-B.

Our visualization tool also allows us to zoom in and examine the detailed connections between roots and CAs. Figure 12 shows an example. Here, the root "Entrust.net Certification Authority (2048)" delegates to six CAs, the largest one being "Entrust Certification Authority - L1C". The certificates issued by this large intermediate CA fall into eight clusters. Upon closer examination (not reflected in the figure) the tool shows that the templates derived from each of these eight clusters indeed have slight differences. With this tool, we can conveniently examine the structure of the clusters and the details of individual CAs and certificates.

Since the identity of the parent CA is a high-weight feature for input, the clustering process naturally factors in the structure of CA hierarchies. However, this feature is not a dominant factor in clustering as we also have many other features about certificate contents (Table V). Still, among our clustering results, we did not observe any cluster spanning across different CAs, suggesting that different CAs may indeed not share the exact same templates.

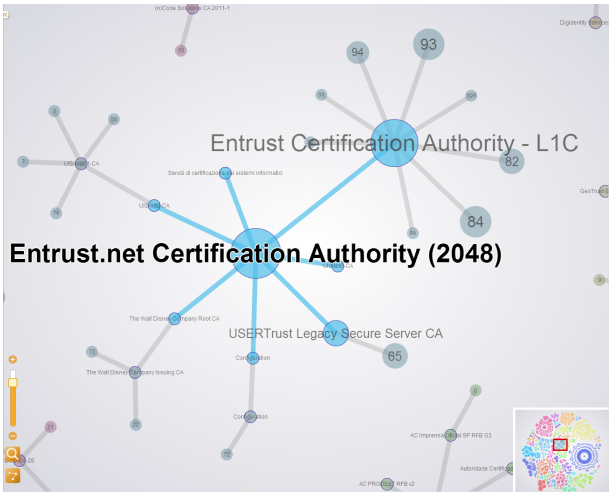

Fig. 12. Zoom on a Root and its Intermediates.

TABLE VI. RECONSTRUCTED TEMPLATE FROM CLUSTERING.

X.509 Fields	
Serial	16.0 bytes entropy avg.
Signature	Sha1-RSA
Subject	CN, OU, O, L, S, C
Validity	14.6 months avg.
Public Key	1952 bits avg.

X.509 Extensions	
Alternative Names	
Basic Constraints	CA=False
Key Usage	Digital Signature Key Encipherment
CRL Points	http://SVRSecure-G3-crl.verisign.com/SVRSecureG3.crl
Policies	2.16.840.1.113733.1.7.54 CPS= https://www.verisign.com/cps
EKU	Server Authentication Client Authentication
AKI	0D445C165344C1827E1D20AB25F40163D8BE79A5
AIA	On-line Certificate Status Protocol http://ocsp.verisign.com Certification Authority Issuer http://SVRSecure-G3-aia.verisign.com/SVRSecureG3.cer

Using the clustering results, we proceed to derive a certificate template for each cluster. Table VI shows an example reconstructed template for the most commonly issued Verisign certificates.

In order to evaluate the relevance of our clustering, we compare the template violations (which directly depend on our clustering features) with the individual certificate violations that we observe on samples from the cluster. In validation of our approach, we find that the affinity to the same cluster is strongly correlated with template violations. In particular, across all clusters with more than 5 certificates, for all rules, in more than 94.5% of instances the fraction of rule violations within a cluster is either all or nothing.

We assign scores to each type of violation based on its likely impact, which we can use to better visualize overall compliance. For instance, missing the CRL distribution points extension is a significant template violation, while listing an expired domain in the subject alternative names is a significant certificate violation. We can then color clusters based on their total template and individual certificate scores. We show the results of this evaluation in Figure 13; the size of small cluster nodes is artificially increased to improve visibility. With our

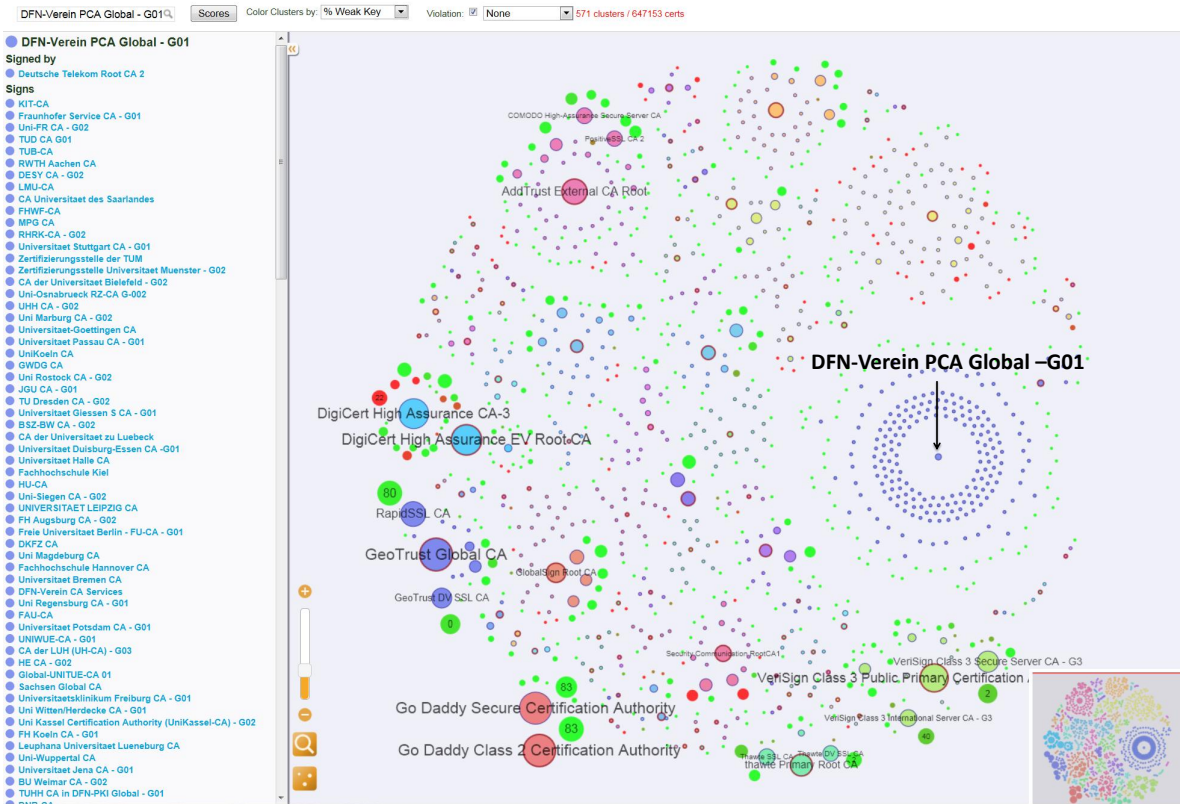

Fig. 11. Distribution of Clusters among CAs. The color scheme reflect the percentage of weak keys in a cluster. The left pane shows the searching interface.

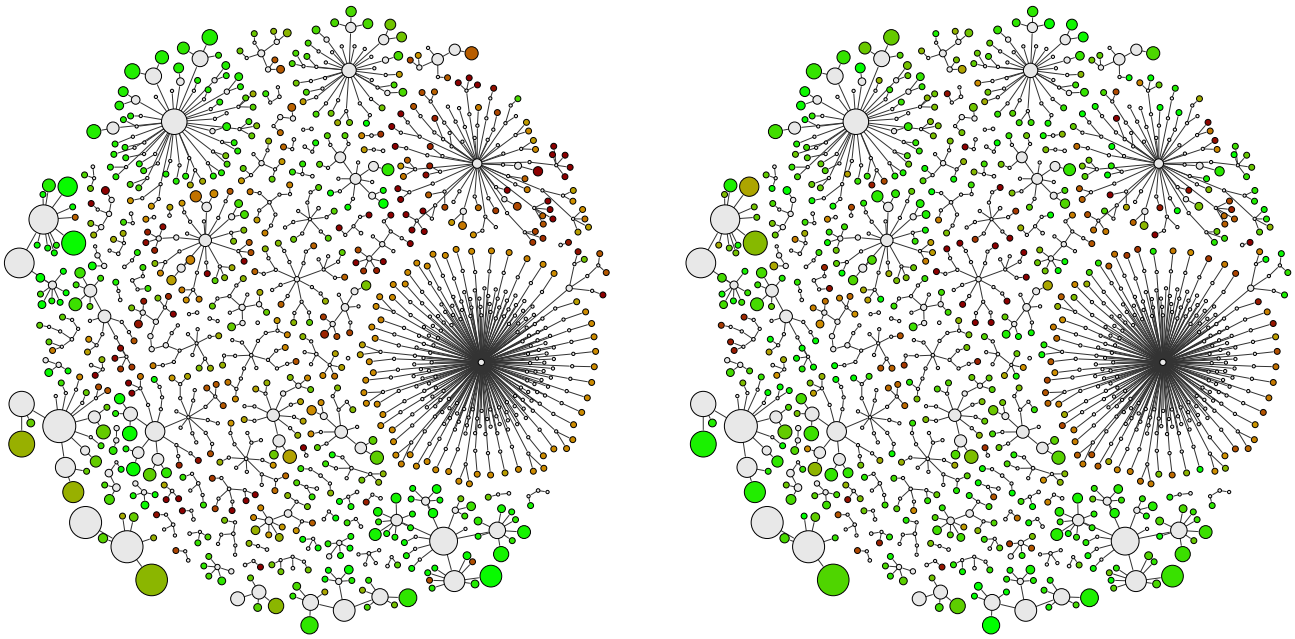

Fig. 13. Comparison of Cluster Quality based on Two Metrics. Clusters are colored based on template scores (left) and average observed violations (right). Clusters are enlarged for better visibility. Green (or low saturation) denotates better compliance.

importance-directed choice of weights, the score correlation between template and individual certificate violations reach 25% on clusters containing more than 50 certificates.

In addition to the global evaluation of the Web PKI, our method could also be used by certificate chain validation software to implement additional checks in high security systems. The clustering information for the entire Web PKI fits in less than a megabyte, and can thus be distributed to clients. Given a certificate, it is easy to find the nearest cluster by comparing the distances with each cluster center of the certificate's issuer. If the distance is too large, or if unusual violations are found, the certificate can be flagged for manual inspection by either by the user or a network administrator. In particular, some of the violations we found have already prompted changes in the validation process in Windows in order to reject certificates that were issued for the Web when used for signing code.

B. Does Size Matter?

Currently the Web PKI has a high degree of concentration: very few CAs issue the vast majority of new certificates, and there is a long tail of smaller authorities. Figure 14 plots the number of certificates signed by each CA in decreasing order (the bold line represents the most recent time period). The top 100 intermediates cover about 98.5% of all certificates for both periods. Thus, the removal of the least used 85% intermediates would impact only 1.5% of websites we connected to.

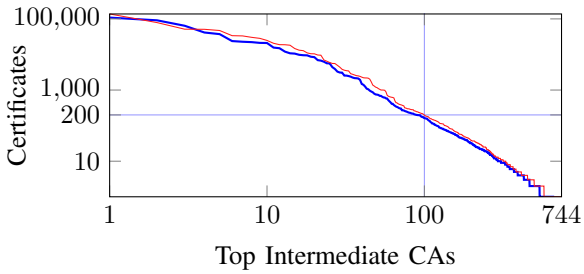

Fig. 14. Number of Certificates by CA.

Whether larger or smaller CAs do a better job policing their certificate issuance infrastructure is open for debate. We find evidence supporting two trends: more delegation is associated with a lesser degree of compliance, and smaller CAs (in particular, those controlled by government entities) tend to exhibit a higher level of violations.

For instance, the CA "DFN-Verein PCA Global - G01" (marked by the arrow in Figure 11) has a large number of intermediates (represented by connected circles) with high scores for both individual and template violations in their clusters. The numerous intermediates correspond to authorities signed to German universities and academic institutions. All together, they represent close to a third of all issuing intermediates, for a total of fewer than 2000 certificates per year.

While the growth of the number of trusted roots is slowing down, as shown in Figure 15, it appears that continued operation of smaller CAs is holding back improvement in the compliance rate.

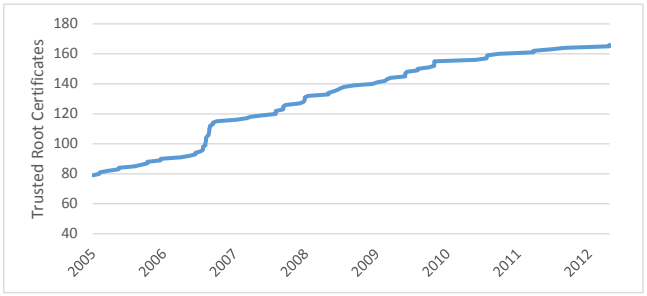

Fig. 15. Growth of the Mozilla Root Program.

Even very compliant root authorities may use a few templates with high violation scores, indicated by dark dots in Figure 13. In some cases, we found obvious mistakes in templates that prompted us to notify the affected CAs directly.

C. DNS Analysis

For individual certificate violations, we perform additional checks that require network queries, as described in §V-C. A DNS query for the Start of Authority (SOA) record of each listed origin allows quick identification of expired domains, and determining whether the origins are served by different DNS servers. Resolving the IP addresses of each domain allows us to check if the server's location matches the subject country listed in the certificate.

For small sample sets, we also look up WHOIS information from the domain registrar, and compare it with the certificate's subject. More importantly, we can detect certificates whose issuance date precedes the entry's creation date, a sign that the owner of one of the applicable domains may have changed. The CA/Browser Forum Requirements mandate control verification of the listed names and IP addresses only at the time of issuance of a certificate, and revocation by the CA when it is explicitly informed that the subject no longer holds control over one of the listed names.

When considering certificates issued after the effective date of the baseline requirements, we find some clusters of domain-validated certificates with over 17% of expired certificates. Our samples also suggest that about 0.5% of certificates valid for two or more years issued in 2011 are for domains that have changed ownership, and we have found a few instances of certificates issued to the new owner. Lingering certificates after a change of domain owner are a major security threat, as they can be used for man-in-the-middle attacks. Public logs of all certificates issued by trusted CAs [5] provide a solution to this problem; however, they are not deployed on a global scale yet.

With the low price of domain validated certificates (comparable to domain registration fees), name squatters may be able to resell their certificates to hackers after selling corresponding domains. Limiting the validity period of a domain-validated certificate to at most the closest expiration date of all the applicable domain names could help mitigate this risk.

D. Content Distribution Networks

We also found large clusters associated with Content Delivery Networks (CDNs) that showed unusual characteristics.

For example, a large CDN cluster associated with GlobalSign OV CA has 2282 certificates, which all use the official policy identifier for organization validated certificates promoted by the CA/Browser Forum in the Baseline Requirements, namely, 2.23.140.1.2.2. These certificates are issued to CloudFlare, Inc. by the GlobalSign Organization Validation CA. Despite their validity period between 4 and 5 years, we found they are being replaced very often.

A CDN is a worldwide distributed network of proxy servers used to speed up access to websites and mitigate denial of service attacks. Some CDN providers offer TLS encryption between clients and their proxy servers (also called points of presence, PoPs). The certificate used for the TLS service can either be provided by the website owner or obtained by the CDN provider from a partner CA. In the latter case, the issuance process is based on DNS delegation to the CDN, without further authorization from the domain owner. However, because PoPs are shared by many customer websites of the CDN, each such certificate is valid for a large set of unrelated domains, which can change frequently. We observe that such promiscuous certificates present several weaknesses.

First, there is no guarantee that the connection between the CDN's PoP and the website's backend server is also encrypted. The absence of encryption is particularly problematic when PoPs are near users but far from backend servers, so communication with those servers may be at higher risk of monitoring. Second, we consider that the CDN is acting as CA by proxy, and the details of the issuance process are not reflected in the certificate's issuance policy. Third, despite the high turnover rate of promiscuous certificates, each time an updated certificate is signed, its previous version is not revoked by the CA. Worse, many domain that are no longer using the CDN service are still listed in promiscuous certificates, sometimes months after the termination of service. We argue that this form of operation should be more strongly regulated, considering the large number of private keys and the delegated signature power granted to CDNs.

E. Entropy Estimation

For a given template, we can also try to estimate whether certificates include the mandatory 20 bits of entropy in serial numbers. This requirement is a cost-effective defense mechanism against collision-finding attacks. Security of X.509 certificates depends critically on the collision-resistance property of the underlying hash function. Collision-resistance of some hash functions (most notably, MD4 and MD5) is manifestly broken, and there are credible cryptanalytic attacks against several others (SHA-1 and GOST 34.11-94).

The most serious scenario of a breach of PKI that relies on attacking the hash function has been described by Stevens *et al.* [37] and deployed in the wild by authors of the Flame malware [1]. In this scenario the attacker submits its certificate request to the CA, and upon obtaining the certificate, replaces its content with another certificate with the same hash. Current technology for forging hash function collisions depends on the attacker's ability to predict or control the initial part of the legitimate certificate. Proactive countermeasures against possible breaches of collision-resistance require CAs to randomize the certificate by generating its serial number (or a portion thereof) at random or by adding randomness to its subject field.

In order to validate compliance with this requirement, we developed an entropy-estimation procedure and apply it at the cluster level. This procedure can produce an *upper* bound on the entropy, since the CA may, for instance, use a pseudo-random function expanding a predictable sequence of inputs (a counter or a timestamp) into a random-looking series. We obtain the estimate by collecting other certificates issued by the same CA, and approximating the average *conditional entropy* of a single certificate from that list, given all other certificates. Concretely, let the sorted list of serial numbers extracted from certificates issued by the CA be $S_1, \ldots, S_n$. The list is sorted according to the certificate's issuance date. For each serial number S_i, we use as an approximation for the S_i's conditional entropy the difference between the compressed length of $S_1|\ldots|S_n$ and $S_1|\ldots|S_{i-1}|S_{i+1}|\ldots|S_n$. The procedure is quite effective in identifying instances where the nominal length of the serial number exceeds its entropy content. Consider the following example of a list of serial numbers:

```
3DAA1A7F000000001CAF   3DFB65A7000000001CBA
3DFB80EF000000001CBB   5B68F796000000001D07
5B6DA3EF000000001D0D   5B70ECB200000004CB9D
5C0F9D92000000001D18   61A57E95000000001D2A
11CD2F73000000001D71   11CD7035000000001D72
11CD9B1E000000001D73   11CDC5A8000000001D74
```

The length of the serial number field is 10 bytes, while our estimate of the conditional entropy is approximately 50 bits per serial number. We run this algorithm on the concatenation of all serial numbers from the cluster to estimate their individual entropy.

We incorporate the results of this evaluation as a template violation if the estimated entropy is less than 20 bits; it was triggered on 2.1% of all clusters, while another 6% had between 20 and 24 bits of entropy in the serial number. For comparison, the serial number of the certificate used for the Flame collision employed a format similar to the one listed above, so it is not clear that 24 bits constitute a sufficient requirement.

VII. Conclusion

In this paper, we have shown considerable evidence that the gap between the PKI guidelines enumerated by the CA/Browser Forum and what exists in practice is in fact shrinking over time. However, these improvements are far from uniform, and in some cases compliance failures raise significant security concerns. Moreover, we point out several instances where the guidelines could be made stronger with significant benefit and little added cost, such as requiring path length and extended key usage constraints in intermediate CAs that sign endpoint certificates.

Our results suggest two important trends with respect to compliance. The majority of larger commercial CAs tends to show adequate adherence to the standards, whereas compliance violations tend to increase with the frequency and depth of authority delegation and the variety of issuance policies exhibited by a given CA. On the other hand, a large number of small corporate and government-operated CAs also show poor compliance.

Finally, we demonstrate and validate a clustering mechanism over collections of certificates that automatically derives templates describing CA behavior. These templates mirror the issuance policy under which certificates were signed, and we use them to drive a visualization interface that can represent the entire publicly visible Web PKI filtered according to specific violations and overall compliance.

We believe our work demonstrates the viability and value of large-scale monitoring of the practices of certification authorities. Beyond our specific results, our study presents a powerful methodology for certificate data analysis on a global scale. Monitoring of this form should be valuable on an on-going basis, both for tracking the adoption of guidelines and more broadly for assessing the health of the Web PKI and identifying specific problems. Our methods not only offer a practical way to keep track of the status of the CA system and discover problems before they are exploited for attacks, but also suggest new signals for evaluating certificate trustworthiness in chain validation software, which we plan to explore in future work.

REFERENCES

[1] J. Ness, "Flame malware collision attack explained," Tech. Rep., Jun. 2012. [Online]. Available: http://blogs.technet.com/b/srd/archive/2012/06/06/more-information-about-the-digital-certificates-used-to-sign-the-flame-malware.aspx

[2] M. Coates. (2013) Revoking trust in two TÜRKTRUST certificates. [Online]. Available: http://blog.mozilla.org/security/2013/01/03/revoking-trust-in-two-turktrust-certficates/

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

[4] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov, "The most dangerous code in the world: Validating SSL certificates in non-browser software," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*.

[5] Google, "Certificate transparency." [Online]. Available: https://sites.google.com/site/certificatetransparency/

[6] OWASP Foundation, "Certificate and public key pinning." [Online]. Available: https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning

[7] Convergence 2011, "An agile, distributed, and secure strategy for replacing certificate authorities." [Online]. Available: http://www.convergence.io/

[8] D. Wendlandt, D. G. Andersen, and A. Perrig, "Perspectives: Improving SSH-style host authentication with multi-path probing." in *USENIX Annual Technical Conference*, 2008, pp. 321–334.

[9] P. Hoffman and J. Schlyter, "RFC 6698—the DNS-based authentication of named entities (DANE)," Tech. Rep., Aug. 2012. [Online]. Available: http://tools.ietf.org/html/rfc6698

[10] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "RFC 4033—DNS security introduction and requirements," Tech. Rep., Mar. 2005. [Online]. Available: http://tools.ietf.org/html/rfc4033

[11] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining your Ps and Qs: detection of widespread weak keys in network devices," in *Proceedings of the 21st USENIX Security Symposium*.

[12] O. Levillain, A. Ébalard, B. Morin, and H. Debar, "One year of SSL Internet measurement," in *Proceedings of the 28th Annual Computer Security Applications Conference*.

[13] 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 2011 ACM SIGCOMM Internet Measurement Conference*.

[14] N. Vratonjic, J. Freudiger, V. Bindschaedler, and J.-P. Hubaux, "The inconvenient truth about web certificates," in *The Workshop on Economics of Information Security (WEIS)*, 2011.

[15] M. A. Mishari, E. D. Cristofaro, K. M. E. Defrawy, and G. Tsudik, "Harvesting SSL certificate data to identify Web-fraud," *I. J. Network Security*, vol. 14, no. 6, pp. 324–338, 2012.

[16] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS certificate ecosystem," in *Proceedings of the 13th Internet Measurement Conference*, Oct. 2013, pp. 291–304.

[17] M. Abadi, A. Birrell, I. Mironov, T. Wobber, and Y. Xie, "Global authentication in an untrustworthy world," in *Proceedings of the 14th USENIX workshop on Hot Topics in Operating Systems*.

[18] Electronic Frontier Foundation, "The EFF SSL Observatory," https://www.eff.org/observatory, 2010.

[19] CA/Browser Forum. (2012, May) Guidelines for the issuance and management of extended validation certificates, v. [Online]. Available: https://www.cabforum.org/Guidelines_v1_4.pdf

[20] Mozilla Foundation. (2012) Mozilla bug 724929: Remove Trustwave certificate(s) from trusted root certificates. [Online]. Available: https://bugzilla.mozilla.org/show_bug.cgi?id=724929

[21] CA/Browser Forum. (2013, May) Baseline requirements for the issuance and management of policy-trusted certificates, v.1.1.5. [Online]. Available: https://www.cabforum.org/Baseline_Requirements_V1_1_5.pdf

[22] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "RFC 5280 - Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," Tech. Rep., May 2008. [Online]. Available: http://tools.ietf.org/html/rfc5280

[23] Microsoft Corporation. (2013) Root certificate program. [Online]. Available: http://technet.microsoft.com/en-us/library/cc751157.aspx

[24] Mozilla Foundation. (2013) CA certificate policy. [Online]. Available: http://www.mozilla.org/projects/security/certs/policy/

[25] Canadian Institute of Chartered Accountants. (2013, Jan.) WebTrust for certification authorities. [Online]. Available: http://www.webtrust.org/homepage-documents/item72056.pdf

[26] European Telecommunications Standards Institute. (2012, Nov.) Policy requirements for certification authorities issuing public key certificates, v2.3.1. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102000_102099/102042/02.03.01_60/ts_102042v020301p.pdf

[27] Debian Security Team, "Cve-2008-0166: PredictableRandom Number Generator," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0166, 2008.

[28] E. B. Barker, D. Johnson, and M. E. Smid, "SP 800-56A. recommendation for pair-wise key establishment schemes using discrete logarithm cryptography (revised)," Gaithersburg, MD, United States, Tech. Rep., 2007.

[29] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide scanning and its security applications," in *Proceedings of the 22nd USENIX Security Symposium*, Aug. 2013, pp. 605–619.

[30] Alexa Internet Inc. (2013) Top 1,000,000 sites (updated daily). [Online]. Available: http://s3.amazonaws.com/alexa-static/top-1m.csv.zip

[31] International Computer Science Institute. (2012) The ICSI certificate notary. [Online]. Available: http://notary.icsi.berkeley.edu/

[32] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 581–590.

[33] U. Kühn, A. Pyshkin, E. Tews, and R.-P. Weinmann, "Variants of Bleichenbacher's low-exponent attack on PKCS#1 RSA signatures," in *Sicherheit*, A. Alkassar and J. H. Siekmann, Eds. GI, 2008, pp. 97–109.

[34] OpenSSL. (2013) Documentation of the X.509 API. [Online]. Available: http://www.openssl.org/docs/apps/x509.html

[35] N. Heninger. (2013) Factoring as a service. [Online]. Available: http://crypto.2013.rump.cr.yp.to/981774ce07e51813fd4466612a78601b.pdf

[36] D. Eastlake, "RFC 6066 - Transport Layer Security (TLS) extensions: Extension definitions," 2011. [Online]. Available: http://tools.ietf.org/html/rfc6066

[37] M. Stevens, A. Sotirov, J. Appelbaum, A. K. Lenstra, D. Molnar, D. A. Osvik, and B. de Weger, "Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate," in *CRYPTO*, S. Halevi, Ed. Springer, 2009, pp. 55–69.