No Attack Necessary: The Surprising Dynamics of SSL Trust Relationships

Bernhard Amann[1], Robin Sommer[1,3], Matthias Vallentin[2], Seth Hall[1]
[1]International Computer Science Institute [2]UC Berkeley
[3]Lawrence Berkeley National Laboratory

ABSTRACT

Much of the Internet's end-to-end security relies on the SSL/TLS protocol along with its underlying X.509 certificate infrastructure. However, the system remains quite brittle due to its liberal delegation of signing authority: a single compromised certification authority undermines trust globally. Several recent high-profile incidents have demonstrated this shortcoming convincingly. Over time, the security community has proposed a number of counter measures to increase the security of the certificate ecosystem; many of these efforts monitor for what they consider tell-tale signs of man-in-the-middle attacks. In this work we set out to understand to which degree *benign* changes to the certificate ecosystem share structural properties with attacks, based on a large-scale data set of more than 17 billion SSL sessions. We find that common intuition falls short in assessing the maliciousness of an unknown certificate, since their typical artifacts routinely occur in benign contexts as well. We also discuss what impact our observations have on proposals aiming to improve the security of the SSL ecosystem.

1. INTRODUCTION

As a key building block of today's Internet security, the Secure Sockets Layer (SSL[1]) protocol provides secure end-to-end channels and authentication through its underlying X.509 certificate infrastructure. In a nutshell, certificate authorities (CAs) sign server certificates, which clients then verify against a list of trusted root CA certificates shipping with their operating system or client software. In most cases, root CAs do not sign server certificates directly, but instead delegate signing authority to intermediate CAs. When validating a certificate, a client attempts to build a valid certificate chain from the server certificate to one of the root certificates it knows, including intermediates as necessary. However, since all root and intermediate CAs share the authority to sign *any* certificate Internet-wide,[2] the global trust in the system breaks with the weakest link: the compromise of a single CA undermines the entire X.509 certificate infrastructure. Consequently, CAs represent an attractive target for attackers: numerous CA compromises [32] and questionable issuing practices [24, 31] have demonstrated that adversaries can obtain rogue certificates for well-known identities to launch *transparent* man-in-the-middle (MITM) attacks where victims do not see a warning because the injected certificate validates correctly.

Over time, the security community has proposed a number of counter measures to increase the security of the certificate ecosystem, including TACK [26], DANE [17], and pinning extensions for HSTS [12]. None of them have yet seen widespread adoption, although Chrome's internal certificate pinning has proven effective in specific scenarios [24]. Many of the existing efforts suggest to monitor for what they consider tell-tale signs of MITM attacks, notifying the user, for example, when encountering certificates not yet seen for a particular target domain [7], or when a certificate's issuer changes to a CA in a different country [36]. We notice, however, a striking gap in virtually all of these proposals: none of them systematically analyzes how often corresponding activity occurs in *benign* circumstances. Doing so however constitutes a crucial step in assessing their efficacy since the setting imposes a classic base-rate fallacy [4]: the probability that users fall victim to an actual MITM attack remains extremely low, and hence even small false positive rates will quickly train them to ignore any security-related warnings.

In this work we set out to understand this effect by pursuing a large-scale study of the *trust graph* induced by the SSL certificate infrastructure. We first survey known MITM attacks and analyze the impact of the deployed malicious certificates on the global trust relationships. We then examine a month's worth of *all* daily changes to the certificate graph for similar patterns. Surprisingly, we find that common intuition falls short in assessing the maliciousness of an unknown certificate, since in practice all such artifacts routinely occur in benign contexts as well. As one example, for 1.3K of the certificates that changed in January 2013, the country of the issuing CA changed. We conclude from our study that global trust relationships—which involve a large number of independent actors driven by different interests and incentives—hardly provide a robust basis for detecting patterns of abuse. Specifically, our results have concrete implications for Certificate Transparency (CT) [22], a recent

[1]We will refer to either SSL or TLS as "SSL."

[2]Individual CAs can be technically constrained but this rarely occurs in practice today (see §3.3).

ACSAC '13 Dec. 9-13, 2013, New Orleans, Louisiana USA

http://dx.doi.org/10.1145/2523649.2523665

proposal to improve the security of the PKI infrastructure.

The key ingredient for conducting this study constitutes a comprehensive data set tracking global certificate changes over an extensive period of time. For more than a year now, we have been collecting SSL certificates (and other session-level SSL features) from upstream network traffic at currently 8 large-scale institutions on an ongoing basis. As of mid September 2013, our data set comprises 1.4M unique certificates, extracted from about 37 billion SSL sessions of more than 314K users in total. The collection provides us with a uniquely broad vantage point for understanding the global SSL ecosystem.

We structure the remainder of this paper as follows: We introduce our measurement infrastructure and data set in §2. After presenting known attacks on the CA ecosystem in §3, we examine daily changes to the global certificate system in §4 in relation to these attacks. We discuss our results in §5, assessing their impact on different ideas and ongoing efforts that aim to improve the security of the SSL ecosystem, and suggesting remedies for some of the newfound problems. After summarizing related work in §6 we conclude in §7.

2. DATA COLLECTION

For more than a year we have been collecting SSL session and certificate information from currently 8 research and university networks, covering activity of approximately 314K active users in total. In this section we describe our collection effort as well as the resulting data set in more detail.

2.1 Setup

All our data providers run the open-source system Bro [33, 6] on their gateway links. We we provide them with a custom Bro analysis script that collects details from each outgoing SSL connection, including its timestamp, certificates, TLS extension information, and more. Every hour, the script uploads ASCII-formatted log files to a database located at our research institute. Due to privacy concerns our script does not record any information that would identify a client system directly. All our data collection sites posses the complete source code of the data collector, and they have used their internal review processes to approve the specifics of the collection.

Our data set exhibits artifacts of the collection process that are beyond our control. As we leverage operational setups that run our analysis on top of their normal duties, we must accept occasional outages, packets drops (e.g., due to CPU overload) and misconfigurations. As such, we deliberately design our data collection as a "best effort" process: we take what we get but generally cannot quantify what we miss. Nonetheless, given the large total volume across the 8 sites, we consider the aggregate as representative of many properties that real-world SSL activity exhibits, including the most commonly seen certificates.

2.2 Data Sets

Table 1 summarizes the data we have collected from each participating site. Our contributors requested to remain anonymous. Most of them represent research environments. Nearly all of them are located in the US (non-US sites include "X" in their labels). As we added the sites incrementally to our effort, the individual sets span different time periods. For comparison, we list the total hours observed at each site (non-continuous due to occasional outages). Two of the sites that originally joined our data collection effort eventually had to leave because of staff changes. As Table 1 shows, our data set contains a total of 57.4M certificates. Of those, over 47 million originate from Grid traffic and Tor servers. Due to the highly dynamic and specialized nature of these two applications, we exclude them from the further discussion in this paper. The *filtered* column in Table 1 shows the number of remaining certificates.

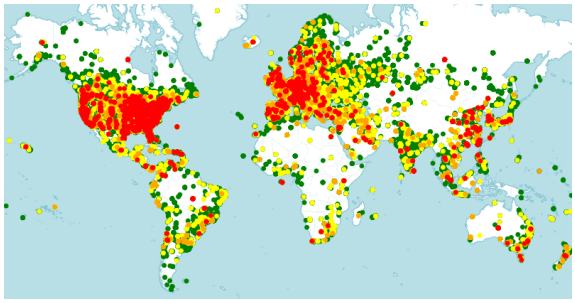

Figure 1: Map view of server IP addresses. Each dot represents one IP. Color encodes number of connections, where green means < 100, yellow > 100, orange $> 1,000$, and red $> 10,000$.

To keep the data analysis manageable, we limit our further discussion to changes in the SSL infrastructure occurring during January 2013. Out of the 842.7K filtered certificates that we had seen in the 17.5B connection observed as of February 1, 2013, 622K were not expired. Of those, we could validate 489.6K against the Mozilla root-store. For certificate validation, we aim to match the results a typical browser would give. To this end, we deploy the NSS library, which Firefox and Chrome use to validate certificates, and retrieve missing intermediate CA certificates using the *Authority Information Access* (AIA) X.509v3 extension.

For the remainder of this paper, we only consider certificates we could validate at the time of the corresponding SSL connection. The connection counts of Table 1 only include successfully established SSL connections, ignoring cases where our monitors reported traffic loss. Even though most of our providers are based in the US, our data exhibits a geographically diverse perspective of SSL servers: Figure 1 shows a map of the server locations based on IP address.

3. ATTACK SURVEY

In this section we survey known attacks and analyze them through the lens of our global *trust graph* in which vertices correspond to certificates and edges to globally valid trust relationships. To set the stage, we begin by briefly summarizing a set of basic properties of the overall graph, and then proceed to examine the specifics of recent high-profile attacks for understanding what facilitated their success.

In our discussion we only consider attacks on CAs pertaining to the web infrastructure while excluding alternate trust hierarchies that also deploy SSL (such as Grids). We neither cover attacks on client/server implementations or on the SSL protocol itself, as our primary focus concerns adversaries launching transparent MITM attacks using a malicious certificate that validates correctly.

3.1 Trust Graph

Our data collection provides us with a comprehensive set of certificates seen "in the wild", which we use to derive a

Site			Certificates		Connections	Time	
Label	Type	Est. Users	Total	Filtered	Total	Hours	Start (– End)
US1	University	90,000	54,883,526	1,064,786	15,029,983,518	13,046	02/12
US2	Research site	250	643,992	46,560	203,095,274	11,383	02/12
US3	Research site	4,000	316,190	150,871	1,330,286,118	12,229	02/12
US4	University	50,000	1,708,874	418,689	7,605,351,160	11,092	02/12
X1	University	3,000	13,798	8,755	10,591,869	3,392	03/12 – 09/12
US5	Gov. Network	50,000	186,928	171,269	787,579,602	7,355	04/12 – 09/13
US6	University	30,000	350,928	196,482	942,039,166	7,790	08/12
US7	University	100,000	835,283	370,775	9,067,412,407	7,904	08/12
US8	Backbone Network [2]	30,000	33,747	32,256	636,405,991	3,282	01/13
X2	University	10,000	127,104	74,227	1,336,627,826	7,426	11/12
All [1]		314,250 [3]	57,359,391	1,384,255	36,949,381,778	—	—

[1] The total reflects the number of *unique* items across all sites.
[2] At the moment only a small fraction of the total backbone traffic is examined, representing about 30,000 users.
[3] Only counting active sites.

Table 1: Summary of data set properties from contributing sites.

Root Certificates			Owners		
%	Root	Owner	%	Roots	Owner
19%	GeoTrust	Symantec	38%	14	Symantec
18%	Go Daddy	GoDaddy	20%	5	GoDaddy
14%	AddTrust	Comodo	16%	4	Comodo
9.6%	GlobalSign	GlobalSign	9.8%	3	GlobalSign
8.6%	VeriSign(1)	Symantec	4.5%	3	DigiCert
6.3%	Thawte	Symantec	2.6 %	3	Entrust
4.4%	DigiCert	DigiCert	1.7%	1	StartCom
4.1%	USERTRUST	Comodo	1.4 %	3	Verizon
2.8%	VeriSign (2)	Symantec	0.78%	2	Trustwave
2.2%	Starfield	GoDaddy	0.47%	1	DTAG

Table 2: Top 10 root-certificates and owners.

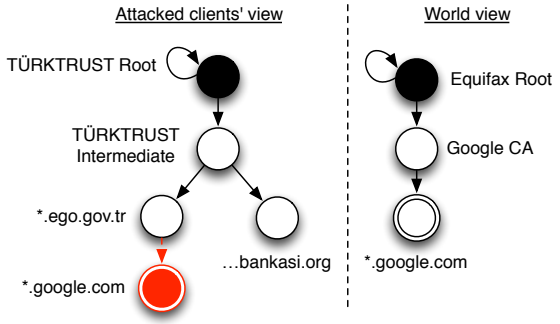

Figure 2: Türktrust attack.

directed graph of global trust relationships. In this *trust graph*, nodes represent certificates of either CAs or end hosts, and incoming edges indicate the CAs that signed them. The trust graph changes over time due to certificate expiration and addition of new certificates.

Table 2 lists the different root certificates we encounter, their owners, and the percentage of the total certificates that we can trace back to them. Symantec is by far the largest CA represented in our data set, having issued 38% of the total certificates under a number of different CA brands, and using 14 different root certificates. The certificates in our data are derived from a total of 84 of the 156 roots included in the Mozilla root store. In total, we see certificates issued by 44 different organizations. Notably, we see certificates signed by government-controlled roots, including Turkey, France, Spain, Hong Kong, the Netherlands, China, and Denmark (which owns the Nationalbank).

3.2 Notable Attacks and Incidents

The last few years have witnessed numerous CA attacks and incidents, each of which involved an adversary attempting to inject a new certificate into the global trust graph such that victims would accept it without facing a warning. To do so, the attackers employed different strategies which we showcase below.

3.2.1 *Türktrust*

At the beginning of 2013, Türktrust accidentally marked two certificates issued to customers as *CA* certificates, which in principle enabled their owners to generate globally valid signatures for any certificate Internet-wide. One of the customers noticed the nature of the certificate at a later point in time and installed it on a network gateway for intercepting encrypted traffic of local users. Google eventually detected the attack when Chrome's certificate pinning reported an unexpected certificate for `*.google.com` [24].

Figure 2 shows the corresponding section of the trust graph for this incident. Google issues certificates for its domains using their own intermediate CA, which is signed by the Equifax Root CA. Our data set includes 33 valid certificates for `*.google.com`, all of them signed by one of their own intermediate CAs.[3] In the Türktrust case, a new `*.google.com` certificate joined the global trust graph as a child of an accidentally created intermediate certificate. This incident stands out as it did not involve an actual attack on the CA but rather an unfortunate mistake. Ultimately, this scenario is indistinguishable from attacking (or coercing) the CA to issue a malicious intermediate certificate.

3.2.2 *Trustwave*

In 2012, TrustWave issued an intermediate CA certificate to one of their customers who then deployed it to transparently decrypt user traffic by generating valid end-host certificates on the fly [31]. The public only became aware of this incident when TrustWave, on their own initiative, revoked the certificate and announced to refrain from issuing such certificates to their customers in the future.

Conceptually, this incident resembles the Türktrust case: a legitimate root issues a new intermediate CA certificate for MITM attacks.

[3]Specific Google services, such as `mail.google.com` and `upload.video.google.com` also use certificates from other issuers.

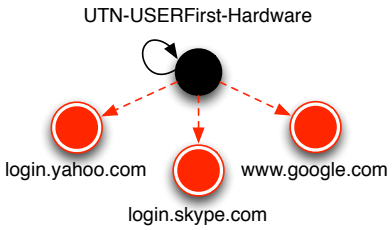

Figure 3: Some certificates from the Comodo attack.

3.2.3 Comodo

In 2011, hackers compromised the Comodo CA and generated 9 illegitimate certificates for well-known web sites, including Google, Yahoo, Mozilla, Skype and Microsoft [9]. Figure 3 shows the corresponding section of the trust graph. The certificates were issued directly from one of the root CAs belonging to Comodo. The attackers created certificates for the common names `mail.google.com`, `www.google.com`, `login.yahoo.com` (3x), `addons.mozilla.com`, `login.live.com` and `globaltrustee`. However, only one certificate for `login.yahoo.com` was encountered in actual use. In the Comodo case, the attackers managed to attack a root CA itself, but were not interested or not able to create new intermediate CA certificates, forcing them to target specific domains.

3.2.4 DigiNotar

In early 2011, a hacker compromised the DigiNotar CA and issued valid certificates for a diverse set of sites, including `*.google.com`, `*.skype.com`, and `*.*.com`, as well as several intermediate CA certificates carrying the names of well-known roots [34]. The `*.google.com` certificate was used to conduct a MITM attack against Internet users in Iran accessing Google services such as GMail. Conceptually, the DigiNotar attack combines the aforementioned incidents: the attackers created both new CA and endhost certificates through an existing CA. From the perspective of the global trust graph, this attack inserts new certificates into the graph, with labels matching existing certificates already associated with different roots.

3.2.5 RapidSSL and Flame

This subsection presents two attacks that are very different from the previously mentioned ones. In 2009, a hacker group performed a proof of concept attack on RapidSSL to demonstrate the problems of using MD5 as a signature algorithm for certificates. The group used a chosen-prefix collision attack to create a rogue intermediate CA certificate that appeared as signed by the RapidSSL CA. They performed this task by creating two certificates with the same MD5 hash value [37]. One of them was a normal end-host certificate, which was submitted and signed by RapidSSL. The second one was a CA certificate (see Figure 4). Due to the fact that both certificates shared the same hash, the signature of the end-host certificate was also valid for the rogue CA certificate. Such a hash collision attack exploits the fact that only the hash of a certificate is signed. If an attacker can create an independent certificate matching a signed hash, that certificate will also validate correctly against the same root. The Flame malware also used this type of attack. Today this type of attack is no longer practical since modern browsers stopped accepting MD5 hashes. The weakest hash algorithm in use today is

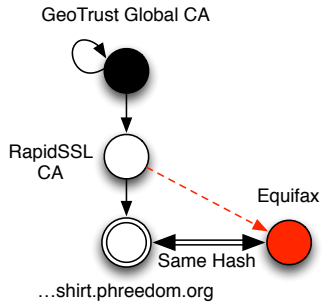

Figure 4: RapidSSL attack.

SHA-1, which the community still considers safe against collision attacks. From the perspective of the trust graph, these attacks exhibit different properties than the former incidents. While once again new certificates join the graph, in this case they come with hash values that match existing ones.

3.2.6 Faults in CA Processes

Numerous incidents exploited deficiencies in a CA's internal processes. For example, Michael Zusman created a certificate for `live.com` by registering the email address `sslcertificates@live.com`. Thawte, a VeriSign intermediate CA, accepted this address as proof of ownership [41, 13]. Another attack relied on including an encoded NUL character (ø) in a certificate's domain name. When using a name like `bank.comøevil.com`, some CAs only validated that the registering user owns `evil.com`. However, some browsers (e.g., Firefox [30]) ignored everything following the NUL character and accepted the certificate for `bank.com` [21, 25]. This bug has been fixed on both browser and CA side. In the context of the trust graph, such cases look similar to the Türktrust incident, where the new certificate relates to already existing domains.

3.3 Remedies

One of the main problems with the current trust ecosystem originates from CAs, including intermediates, who have the ability to issue certificates for any domain. In principle, the X.509 *name constraint* extension should prevent this by limiting intermediates down the chain to issuing certificates for restricted domains only (e.g., subdomains their customers own). However, at the moment the use of name constraints in certificates is still extremely rare. The main reason boils down to lack of support in current browsers; Safari and iOS in particular do not yet honor this extension.

Extended validation (EV) certificates represent another effort to increase trust in the CA system. However, their actual benefit remains unclear because users often cannot differentiate the certificate types. Furthermore, SSL rebinding attacks [38, 20, 35, 5] can circumvent EV protection. Finally, EV certificates only indicate that the CA uses stricter standards when checking a customer's identity—which is of no use if the CA has been compromised. For example, the compromised DigiNotar CA was approved for EV by Mozilla [28]; Türktrust received Mozilla approval for EV just before their incident occurred [29].

4. STRUCTURAL EXPLORATION

We continue with an analysis of the trust graph to identify benign changes that structurally resemble attacks, and thus

might be mistaken for malicious activity in the absence of any further context. For this discussion we assume that our data does not contain any MITM attacks. We specifically searched for the fraudulent certificates discussed in §3.2 (e.g., the intermediates that Türktrust accidentally issued), yet did not find any of them in our data set. More generally, the absolute number of Internet-wide MITM attacks is presumably small and dwarfed by the total number of sessions included in our data set, which renders the chance of having recorded an actual attack negligible. In addition, we manually investigated all the specific cases we report and indeed deem them benign.

4.1 Terminology

We define the first appearance of a new certificate as a *change* of the trust graph. Two or more certificates *match* if they share at least one label. The *neighborhood* of a new certificate constitutes the set of certificates that it matches. When a new certificate appears for the first time, we classify the nature of the change by computing a change vector of its key features, including:

1. Size of the neighborhood
2. Number of labels in the new certificate
3. Total number of unique intermediates across all existing certificates
4. Percentage of matching intermediates (*intermediate weight I*)
5. Number of unique roots for the existing certificates
6. Percentage of matching roots (*root weight R*)
7. Time difference between when a certificate became valid and when we first saw it
8. Minimum, maximum, and average overlap between the validity periods of new and existing certificates
9. Minimum, maximum, and average difference when we first encountered the new and the existing certificates
10. Number of different keys among new and existing certificates.

We leverage these vectors for grouping changes that exhibit similar characteristics, and for identifying examples to present in our discussion. While the remainder of this paper focuses on individual observations, we note that for features that we do not discuss further, we were not able to discern stable patterns that might indicate certificate attacks.

The two most important metrics concern the intermediate weight I and the root weight R. For a given certificate, I represents the percentage of its neighborhood with the same issuer. For example, $I = 1$ means that the issuer of the new certificate matches all existing intermediates, and $I = 0.5$ that the issuer of the new certificate matches half the certificates in the neighborhood. We define R correspondingly to represent the percentage of a certificate's neighborhood with the same root CA.

4.2 Data Overview

At the beginning of January 2013, we have seen 741,424 certificates in total, out of which 489,551 are still valid and thus part of the trust graph at this time. Through January, we encounter 80,466 unique new certificates, of which 54,321 validate against the Mozilla root store; we examine the latter subset in the following. 40,885 of the those changes do not affect any other certificate, i.e., the addition either applies to labels that we have not seen at all yet, or the existing certificates have already expired. The certificates that we encounter for the first time lead back to 337 different issuers at 80 different roots.

Looking at the new certificates matching existing ones, we see that most exhibit a small neighborhood, with 9,400 of them matching exactly one certificate in the graph. Most of those certificates replace a certificate that expires soon. However, we also find certificates with rather large neighborhoods: 1,382 larger than 20, and 224 larger than 100. As we will see in the next sections, these often belong to CDNs and big hosting providers. The certificate with the most extensive neighborhood matches 657 existing certificates and belongs to Google.

4.3 Inconsistent Neighborhoods

In the cases discussed in §3.2, the malicious certificates were issued by a different CA that had not signed the benign certificate in the past. Conceptually, this kind of attack splits into two cases: either a previously unknown intermediate CA signs the malicious certificate (as in the Türktrust and Trustwave incidents), or an already established CA (e.g., Comodo) does. In both cases we see a "hand-over" from an existing set of CAs to one not previously seen for the domain. In theory, a hand-over happens either when a certificate changes its intermediate but remains rooted in the same sub-tree in the trust graph, or when a certificate migrates to a new root. In all previously examined attacks, the latter scenario occurred.

To find CA hand-overs, we examine issuers changing between certificates that are valid for the same domain, using the previously introduced metrics I and R. When looking at certificates that joined the graph in January, we find 3,051 for $I = 0$, 8.6K for $I = 1$, and 1,766 for $0 < I < 1$. Hence, a large number of certificates share the same issuer as previous certificates, but there exists also a significant number where this is not the case. We find a similar situation for R: 2,507 for $R = 0$, 9,191 certificates for $R = 1$, and 1,738 for $0 < R < 1$ for.

4.3.1 Large Neighborhoods

As a first step, we examine new certificates that join the trust graph, have a neighborhood of more than 20 certificates, and trace back to a different root than their neighborhood ($R = 0$). We find several certificates where the neighborhood contains as many as 13 different roots. Those all belong to `www.yottaa.net`, a web-optimization CDN, whose certificates cover numerous different hostnames (30-34) and have a neighborhood size of 54-57. In addition to the servers provided by Yottaa, some of their clients also use their own servers where they host certificates for domains which also occur in Yottaa's certificates.

For certificates with slightly fewer roots (more than 6) we encounter a larger variety of CDNs. For example, Incapsula uses a number of certificates with hostnames that we can trace back to 9 roots, while we find 7 roots for Cloudflare. Yottaa, Incapsula, and Cloudflare, participate with 38, 132, and 2.8K certificate domains in the Alexa top-million list, respectively. This list contains several attractive targets for MITM attacks. Consider `foursquare.com`, for example, whose CA structure we show in Figure 5. When the Cloudflare certificates joined the trust graph (at the end of December and the beginning of January) we already knew other certificates for `*.foursquare.com` issued under two different roots. A human observer may

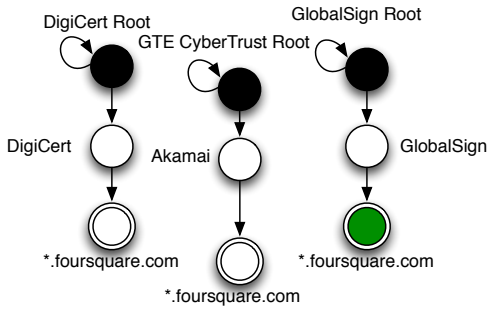

Figure 5: `*.foursquare.com` with new Cloudflare cert.

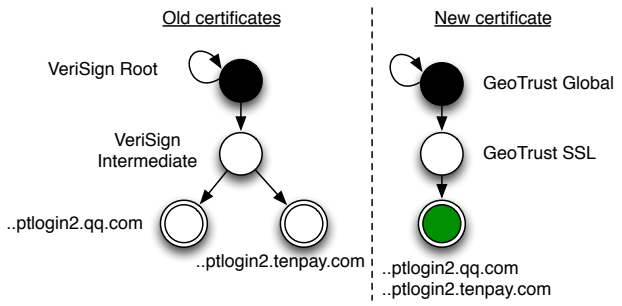

Figure 6: `qq.com` trust graph change.

conclude that this represents a benign change by having learned that Cloudflare operates as CDN and tends to use GlobalSign as CA. However, when looking at the change without such context, it may raise suspicion to see a new certificate from an unrelated root.

Varagesale represents another case that demonstrates this challenge: `varagesale.com` used Cloudflare as CDN provider until mid-January, when we observed a new SSL certificate for the same domain. At this time, we have already recorded 30 other certificates for `varagesale.com`, each of which is used by Cloudflare, issued by GlobalSign, remains valid, and sometimes has been seen months ago already. However, the new certificate was issued by a different root (Comodo) than all the other 30. In cases like this we deem it impossible to separate benign changes from attacks without further context.

Note that Akamai, a well-known CDN, operates differently. In our January change set, we see 104 new certificates that trace back to Akamai. Sites using Akamai can either choose if they want to host content on their own domain, or use one of a small set of Akamai domain names. When sites use their own, in contrast to the CDNs mentioned above, Akamai seems to use an individual SSL certificate for each of the sites. Different CAs issue these certificates, with some of them signed directly by the Akamai intermediate CA and the others by either VeriSign or Comodo. Akamai also serves a large number of its customers via a single SSL certificate valid for `a248.e.akamai.net`, `*.akamaihd.net`, and `*.akamaihd-staging.net`. Incidentally, we see this certificate used by the largest number of unique IP addresses (68,794). It is apparently used when customers choose to only *embed* elements into their homepage. Facebook, for example, uses this approach to serve their profile pictures via Akamai. Amazon Cloudfront appears to use a similar strategy, as we observe 9,667 IP addresses serving a single certificate.

4.3.2 Small Neighborhoods

As we generally find CDNs responsible for a significant share of the non-obvious effects, we now specifically examine changes for domains that are *not* using any CDNs, in the hope to find more regularity there. However, our analysis quickly reveals several examples that might appear malicious to an observer. Figure 6 shows an example involving two high-profile domains: `qq.com`, a popular instant messaging service; and `tenpay.com`, a payment service. Both are based in China and owned by the same parent company. According to Alexa, `qq.com` ranks as the 8th most popular site on the Internet (tenpay ranks 774th). At the end of January, we see a single new certificate showing up that is valid for *both* domains,[4] signed by a not-previously seen CA. At this time the original certificates for both sites were well established (we first saw them in February and March 2012) and still valid until mid-December. One could easily mistake this change for an attack. Even if one knew that the names listed in the certificate belong to the same parent company, an adversary could just as well have deliberately choosen them to appear more legitimate. The only way to be sure the new certificate is indeed benign is to ask the domain.

A similar change occurs for the domain `*.americanexpress.com`—likewise a site that makes an interesting target for attackers. Akamai's intermediate CA issued the established certificate for this site; we saw it first in May 2012, only hours after its validity period starts, and it remained valid until May 2013. However, on January 23, we see a new, VeriSign-signed wildcard certificate for the same domain, with a validity period beginning on the 13th of December 2012. No other VeriSign certificate appears in our data set for that domain. Both certificates were then used simultaneously for one more month; after that, the Akamai certificate was apparently phased out.

When going through the list of domains for which we see new certificates in January, we find a number of further changes that likewise exhibit similarities with recent attacks. For example, several other banking sites switch their certificate issuers (including the Bank of India, the first Montana Bank, the Mechanics Bank and the Danish Arbejdernes Landsbank). Only for some of them the old certificates expire around the saem time.

Furthermore, some sites change from well-known to smaller CAs. For example, `iesabroad.com` exhibits a newer certificate from AlphaSSL in addition to several active certificates issued by RapidSSL. We see one of the old certificates in use along with the new AlphaSSL certificate. To an outsider observer this change might look similar to a MITM attack involving a small rogue CA certificate.

4.3.3 Country Changes

To identify malicious CA changes, one concrete recommendation involves monitoring their countries [36], under the assumption that a site rarely switches to a CA in a different country for benign reasons. However, when examining our data in this regard, we see precisely this scenario occurring for 1.3K changes in January: the country code of the CA that

[4]It is also valid for other domains that seem to belong to the same parent company as well; they use the same DNS servers. However, they have different company names and addresses in the whois service.

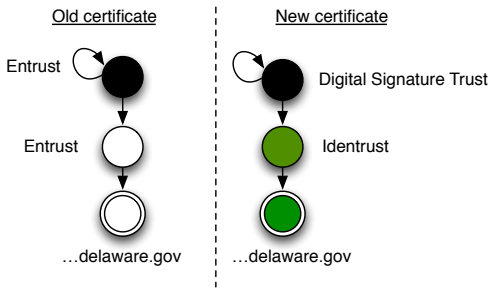

Figure 7: `delaware.gov` trust change with new intermediate CA certificate. New certificates marked.

issues the new certificate is not already part of the existing neighborhood. When looking at root certificates, this change is even more pronounced: 1.7K changes trace back to a root in a different country.

A number of specific cases might look particularly surprising to a human observer. For example, 46 certificates switch to a root located in Israel, including the Nova Scotia Department of Education (`*.ednet.ns.ca`), and `www.privacybox.de`, a German service for anonymously exchanging messages between journalists. We assume that these changes represent benign business decisions to migrate certificates to StartCOM, an CA based in Israel.

Furthermore, `www.zekur.nl` changes from an US-based root to Bermuda (QuoVadis) according to its country code. We assume that many customers are not even aware in which country a service resides; QuoVadis has representations in the UK, Holland and Switzerland besides its Bermuda headquarters—and also operates under those country level domains.

Finally, we note that due to a series of acquisitions and mergers, country codes in certificates often do no longer align with reality. For example, Thawte was based in South Africa before being bought by VeriSign.

4.4 New Intermediate CAs

After having inspected the cases of new end-host certificates, we now turn to intermediate CAs. In our data we find 13 intermediates for the first time in January along with 135 new certificates they have issued, of which four match other certificates in the trust graph. The affected servers include ones for `delaware.gov`, `www.elephanttour.com`, and `www.ph-karlsruhe.de`) (a German high school); as well as a server of the Norwegian DNB finance group switching from a UserTrust certificate to its own intermediate CA. Further analysis of the chains shows that none the new CAs belong to any of the large, well-known CAs, but instead stem from IdenTrust, NetLock (Hungary), Izpene (Spain) and to several members of the German Research Network. Of the 135 new certificates, 120 originate from Servision, a single new Japanese intermediate CA which has a valid CA certificate since November 2012.

In conclusion we regularly encounter new intermediate CA certificates. Furthermore, we see some matches for certificates being issued by these CAs. For outside observers these cases look very similar to the Türktrust/TrustWave incidents, and they could hence easily misinterpret them as attacks.

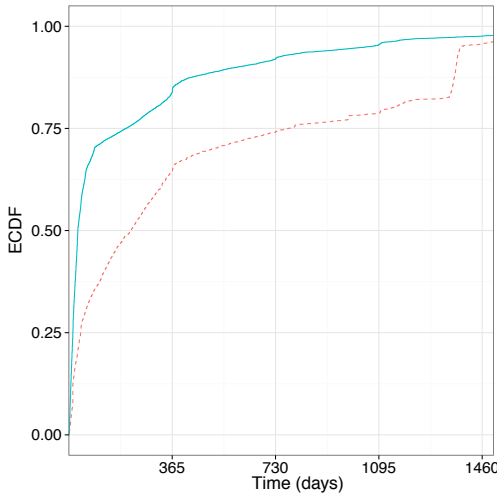

Figure 8: ECDF of minimum validity overlap for a domain (long tail cut). The solid line represents certificates with one other certificate in their neighborhood, the dashed line those with more than one.

4.5 Validity Overlaps

As mentioned earlier, we often see a significant overlap in the validity periods between a domain's certificates. While we expect to see this effect for large sites with data centers spread over the world, or generally sites employing load-balancing techniques, we still find the scale at which overlaps occur surprising. Figure 8 shows the empirical CDF of the validity overlap between a certificate and its neighborhood at the time we first encounter it. [5] The solid line represents certificates with one other certificate in their neighborhood, the dashed line those with more than one. We see a high number of certificates with a small overlap, which one would expect in the simple case where a new certificate replaces an existing one near expiry. However, we also find a sizeable number of certificates with validity periods overlapping by a significant amount of time, both for certificates with a sizable neighborhood (mostly CDNs and large companies like Google) as well as for small sites. The spike of the dotted line between 1,000 and 1,500 days is caused by CloudFlare, which seems to regularly issue new certificates with similar validity periods. For the solid line, there are several small spikes at the one, two, and three year marks, the first being the most significant. Manually examining the certificates, we could not find a discernible reason for overlaps clustered at year boundaries.

4.6 Key Sharing

For domains with many certificates, we frequently see public key reuse. Of all 4,036 new certificates in January that already have more than one matching certificate in the graph, 2,183 share a key with at least one other certificate. Figure 9 shows a comparison between the size of a neighborhood of a certificate and the number of distinct keys that we see in there. For example, the certificate marked by the two dotted lines has a neighborhood of size 93, with each certificate having a different, unique key. It belongs to a web-hosting service, `sureserver.com`. While they use dedicated SSL certificates for each of their servers, they all also share a key

[5] When we know more than one other certificate for a domain in the trust graph, we use the minimal overlap.

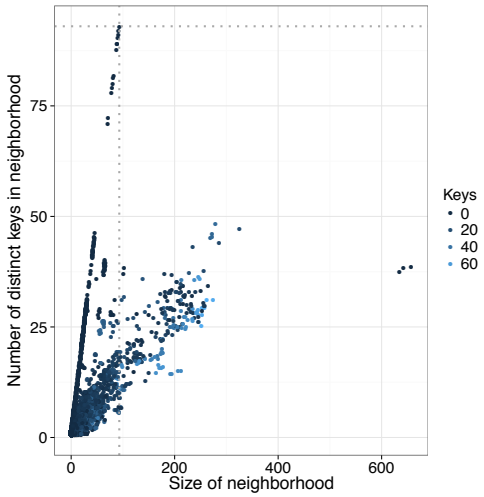

Figure 9: Number of keys in the certificates versus numbers of certificates seen for a label.

with `sureserver.com` itself.

All certificates on the line extending from the origin to this certificate share the characteristic that all members of their neighborhoods have their own unique keys. However, for all certificates to the right of this line, one or more certificates in their neighborhood share the key. The three certificates on the very right of the graph, with over 600 known certificates and about 40 known keys, belong to Google. Indeed, for the 5,409 certificates that we see from the Google intermediate CA, we only observe 206 different keys. Their most common key is shared between 230 certificates. All Google certificates that share keys will expire at exactly the same second on June 7, 2013, but do not exhibit a uniform start validity unlike other certificates sharing a key: they differ by a few minutes , presumably because the Google intermediate CA signed all of them in short sequence. The start times between the different sets of certificates vary wildly, with no clear pattern, such as a correlation between IP ranges and keys. There also are IPs that serve different certificates with different keys.

The color of the individual dots in Figure 9 shows the number of certificates in the neighborhood that share a key with the new one. As the color coding indicates, having a large neighborhood size does not propagate to new certificates. For example, we encounter a new certificate for `*.fiftyflowers.com` while already knowing 60 other certificates for that domain, all at least 5 months from expiry. FiftyFlowers is again hosted by Cloudflare and the 60 certificates have 5 different keys. Besides the Cloudflare certificates, one of the certificates is issued by the UTN-USERFirst-Hardware CA (owned by Comodo). The new certificate has a different key and is issued by GoDaddy. There are other cases like this, and hence we conclude that key sharing does not generalize to not yet seen certificates.

5. DISCUSSION

We now discuss the impact of our results on efforts to increase the security of the CA ecosystem. Our main focus here concerns the challenges that the frequent benign changes impose on approaches aiming to exploit structural properties.

5.1 User-Side Change Monitoring

A number of efforts aim to improve SSL security by comparing server certificates against records of what a browser received in the past [7]. However, the volume and diversity of changes that we observe suggests that any such approach will frequently need to fall back to the user to decide whether proceeding is safe. Unfortunately, for many of the changes that we encounter doing so will not constitute a promising path, as often not even experts will be able to distinguish malicious from benign certificates. Given that actual attacks remain rare, users will quickly learn to click-through any warnings, just as they do today. Soghoian and Stamm [36] present a similar argument, yet proceed by suggesting that warning the user just for CA country changes might strike an acceptable balance. However, our analysis shows that that even these occur much more frequently than the authors seem to expect.

5.2 Certificate Transparency

Certificate Transparency (CT) represents a more promising proposal to improve the current state by generating accountability for CAs. CT aims to thwart MITM attacks by creating a publicly accessible, append-only log of all existing certificates in the Internet. Users or the issuing CA submit new certificates to the log, which records the addition by creating a new signature. This signature (either embedded in the certificate or sent by the server through a TLS extension) proves presence of a certificate in the global log. Everyone can monitor the log for malicious changes and directly notify site operators and CAs. While today there is no way for the public to know to which sites CAs have issued certificates, CT will force them to publish that information in a set of public, audible logs; clients will eventually reject certificates that they cannot find there.

Conceptually, our data set provides a similar global perspective of the certificate ecosystem as CT will once it becomes operational. While our collection lacks the authoritativeness of CAs directly providing input, it nevertheless allows to understand the challenges of monitoring changes to the trust graph by independent 3rd parties. In other words, organizations monitoring CT will encounter similar effects as we do in this study.

One of the main motivations for CT concerns its ability to detect fraudulent certificates for domains that get added to the public logs in preparation for a MITM attack. For large corporations like Google or Facebook, this indeed solves the problem as they will have the resources to monitor the logs continuously and react swiftly to any unauthorized certificates that might appear for their domains. In principle, any other domain owner could do the same. However, we believe that in practice many smaller sites will lack the capabilities, expertise, and probably also the incentive to watch CT on a ongoing basis. With that, it will be left to external parties to monitor the public logs for suspicious changes. These however will face just the same ambiguous situations that our discussion in the previous section highlights. Indeed, not even CAs can take the role of CT monitors as they typically will not know further certificates that their customers might have purchased from the competition. On top of that, it seems plausible to assume that they do not have much of an interest in taking on such a role as otherwise they could have long devised an information sharing initiative between themselves.

To the best of our knowledge, this aspect of CT has not yet received much attention. The CT RFC draft states that "the logs do not themselves detect misissued certificates, they

rely instead on interested parties, such as domain owners, to monitor them and take corrective action when a misissue is detected" [22]. While CT will clearly present an immense step forward for protecting today's fragile trust relationships, it will not provide a silver bullet.

5.3 Possible Remedies

There are several possible remedies to the problems that we point out in this section. Assuming CT gets adopted, there are several easy ways in which the current actors of the system could make it safer. CAs could leverage CT to improve the trust into the certificate systems. For example, a CA could search the logs for already existing certificates before issuing a new certificate for a domain. If it finds existing certificates, it can ask for proof that the private keys indeed belong to the requester. This could potentially eliminate attacks of the type mentioned in §3.2.6, where a CA issued a `live.com` certificate to a non-authorized user.

Also, we can imagine 3rd-party services emerging that interact directly with domain owners. Rather than all web sites directly following the CT logs, they would contract an external entity, providing it with regular updates on legitimate certificate changes regarding their domain. These services would thus know what to watch for. Still, this approach requires an awareness of the problem space on the side of the server operator, as well as a financial incentive to enter such a relationship.

Another mechanism to allow external parties to assess new certificates could involve an X.509 extension to prove that the certificate creator possesses the keys for the domain's existing certificates. This approach should work well for smaller sites as typically they just would have to include proof for their single existing key. While this approach poses more implementation challenges particularly for CDNs, it might work well in combination with CT. CDNs might use alternative, more complex methods like additionally pinning certificate keys or allowable CAs to their domain using DANE [17], TACK [26] or other pinning proposals [12]. This would significantly raise the bar for attackers on the system and allow CAs and external entities to verify that new certificate (requests) are indeed legitimate.

6. RELATED WORK

Studies. The Electronic Frontier Foundation (EFF) popularized the study of X.509 certificate infrastructure by publishing the results of active scans of the full IPv4 address space in 2010 [11]. Holz et al. [18] compare the 2010 EFF data set with active and passive measurements of their own. Their two passive data sets span periods of approximately two weeks. Vratonjic et al. [39] and Mishari et al. [27] study X.509 certificates from the Alexa Top 1 million list, and from randomly scanning domains respectively. Heninger et al. [15] present a weak key study of TLS and SSH keys retrieved by an IPv4 address space scan. Devdatta el al. [2] present a study of SSL error codes and their reasons on the web. Durumeric et al. [10] present a study of SSL certificates retrieved by 110 scans of the IPv4 address space. To the best of our knowledge, our SSL monitoring effort is the only effort that is continually monitoring a significant part of the global SSL landscape, and can thus facilitate a fine-granular change classification.

Deep Infrastructure Changes. The community proposed several new standards and ideas to increase the security of SSL. The DNS-based Authentication of Named Entities (DANE) [17] RFC proposes to embed certificate information into DNS using DNSSEC. DANE can either replace or complement the current CA system, providing a secondary trust anchor. DNS Certification Authority Authorization (CAA) [14] has a similar scope and goal. The Trust Assertions for Certificate Keys (TACK) [26] standard proposal aims to reduce the dependency on CA providers by creating a separate PKI layer to only sign the public keys of servers. TACK operates in a trust-on-first-use (TOFU) mode, where a client connecting to a server initially relies on the existing PKI to validate the server's certificate. In the response, the server also sends its TACK public keys. For subsequent connections, the TACK keys are used to verify the server's public key directly.[6] The IETF standard proposal Certificate Transparency (CT) [22] has received a tremendous amount of support from the community; see §5.2 for more. Clark et al. [8] systemize and evaluate these and several further approaches. While potentially eliminating some of the problems of the global CA system, wide-scale adoption of these proposals seems generally unlikely, due to the required deep client-side and server-side changes. For example, Chrome was the only major browser which supported DANE for a while, but has dropped support in recent versions.

Client-Side. Soghoian and Stamm [36] present the threat of compelled certificate creation attacks, in which governments may force a CA under their jurisdiction to issue malicious certificates for MITM attacks. The authors evaluate several theoretical scenarios in which such man in the middle attacks might be carried out and propose to solve the problem by displaying a warning if the CA is situated in a different country than the entity for which the certificate was generated. Similar in intention but wider in scope, the Firefox extension Certificate Patrol [7] records certificate information for all websites that a user visits. The extension alerts the user when, on a later visit, the site certificate has changed. The user can then examine the certificate change and decide to accept or reject the new certificate. Google introduced hard-coded certificate pinning to Chrome [23] for a certain subset of certificates belonging to Google and large sites. This approach requires no server-side changes, but neither scales nor allows users to modify the internal certificate list.

Server-Side. HTTP Strict Transport Security [16] allows a site to specify that it is only accessible using HTTPS. When a site uses the extensions, supporting browsers will refuse clear-text connections and those not presenting a valid, non-selfsigned certificate chain in the future. A proposed pinning extension to HSTS [12] has similar intentions to TACK and allows a server to tie certificate keys to its domains.

Notaries. Notaries represent an alternative approach to improve the existing state without architectural changes by maintaining a third-party database of server certificates and/or connecting to the server from different parts of the Internet. When clients encounter a certificate, they can match it against the notary's version. Perspectives [40] pioneered this method and Convergence [1] provides an improved implementation. Similar in intention, but for research purposes, Crossbear [19] operates as a MITM origin detector which uses distributed sensors to pinpoint the attack location.

The ICSI SSL Notary [3] is our own DNS-based notary

[6]Theoretically TACK keys can also be shared between clients. However, this needs a separate trusted infrastructure.

service, which makes a subset of the data set used in this paper available to the public. It allows to query when and how often our data providers have encountered specific certificates.

7. CONCLUSION

Certificate changes prove frequent and manifold within the SSL ecosystem. We compare routine changes seen throughout the global certificate trust graph with recent attacks, and we find the two to share many properties, including some that have previously been proposed to separate benign from malicious certificates. We discuss a range of examples that we discover in an extensive data set collected over about a year at the border gateways of 8 large-scale institutions, totaling about 17 billion SSL sessions. Generally, we conclude that without further context it remains impractical to identify malicious certificates from structural properties alone.

8. ACKNOWLEDGMENTS

This research was supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD); by the National Science Foundation under grant number ACI-1032889; and by the U.S. Army Research Laboratory and the U.S. Army Research Office under MURI grant number W911NF-09-1-0553. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DAAD, the NSF, the ARL, or the ARO.

9. REFERENCES

[1] Convergence. http://www.convergence.io.

[2] Akhawe, D., Amann, B., Vallentin, M., and Sommer, R. Here's My Cert, So Trust Me, Maybe? Understanding TLS Errors on the Web.

[3] Amann, B., Vallentin, M., Hall, S., and Sommer, R. Extracting Certificates from Live Traffic: A Near Real-Time SSL Notary Service. Tech. Rep. TR-12-014, International Computer Science Institute, Nov. 2012.

[4] Axelsson, S. The Base-Rate Fallacy and Its Implications for the Difficulty of Intrusion Detection. In *ACM CCS* (1999).

[5] Biddle, R., van Oorschot, P. C., Patrick, A. S., Sobey, J., and Whalen, T. Browser Interfaces and Extended Validation SSL Certificates: An Empirical Study. In *ACM CCSW* (2009).

[6] Bro NSM. http://www.bro.org.

[7] Certificate Patrol. http://patrol.psyced.org/.

[8] Clark, J., and van Oorshot, P. C. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. In *IEEE Security and Privacy* (2013).

[9] Comodo Report of Incident on 15-MAR-2011. http://comodo.com/Comodo-Fraud-Incident-2011-03-23.html.

[10] Durumeric, Z., Kasten, J., Bailey, M., and Halderman, J. A. Analysis of the HTTPS Certificate Ecosystem. In *Internet Measurement Conference* (2013).

[11] EFF. SSL Observatory. https://www.eff.org/observatory.

[12] Evans, C., and Palmer, C. Certificate Pinning Extension for HSTS. IETF Internet-Draft, Sept. 2011.

[13] Goodin, D. How is SSL hopelessly broken? Let us count the ways, 2011. http://www.theregister.co.uk/2011/04/11/state_of_ssl_analysis/.

[14] Hallam-Baker, P., Stradling, R., and Laurie, B. DNS Certification Authority Authorization (CAA) Resource Record. IETF Internet-Draft, May 2011.

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

[16] Hodges, J., Jackson, C., and Barth, A. HTTP Strict Transport Security (HSTS). RFC 6797, Nov. 2012.

[17] Hoffman, P., and Schlyter, J. The DNS-Based Authentication of Named Entities (DANE): TLS Protocol. RFC 6698, Aug. 2012.

[18] Holz, R., Braun, L., Kammenhuber, N., and Carle, G. The SSL Landscape: A Thorough Analysis of the X.509 PKI Using Active and Passive Measurements. In *ACM SIGCOMM* (2011).

[19] Holz, R., Riedermaier, T., Kammenhuber, N., and Carle, G. X.509 Forensics: Detecting and Localising the SSL/TLS Men-in-the-Middle. In *ESORICS* (2012).

[20] Jackson, C., Simon, D. R., Tan, D. S., and Barth, A. An Evaluation of Extended Validation and Picture-in-Picture Phishing Attacks. In *USEC* (2007).

[21] Kaminsky, D., Patterson, M. L., and Sassaman, L. PKI Layer Cake: New Collision Attacks against the Global X.509 Infrastructure. In *Financial Cryptography* (2010).

[22] Langley, A. ImperialViolet – Certificate Transparency, Nov. 2011. http://imperialviolet.org/2011/11/29/certtransparency.html.

[23] Langley, A. ImperialViolet – Public key pinning, May 2011. http://imperialviolet.org/2011/05/04/pinning.html.

[24] Langley, A. Google Online Security Blog – Enhancind Digital Certificate Security, Jan. 2013. http://googleonlinesecurity.blogspot.com/2013/01.

[25] Marlinspike, M. More Tricks for Defeating SSL in Practice. Black Hat USA Talk, 2009.

[26] Marlinspike, M., and Perrin, T. Trust Assertions for Certificate Keys. IETF Internet-Draft, Jan. 2013.

[27] Mishari, M. A., Cristofaro, E. D., Defrawy, K. M. E., and Tsudik, G. Harvesting SSL Certificate Data to Mitigate Web-Fraud. *CoRR abs/0909.3688* (2009).

[28] Mozilla Bug 369357 – Add DigiNotar EV Root CA Certificates, 2012. https://bugzil.la/369357.

[29] Mozilla Bug 433845 – Add TÜRKTRUST Root CA, 2012. https://bugzil.la/433845.

[30] Mozilla Bug 724929 – Improper Character Escaping and Unescaping in alg1485.c & secname.c, 2012. https://bugzil.la/480509.

[31] Mozilla Bug 724929 – Remove Trustwave Certificate(s) from Trusted Root Certificates, 2012. https://bugzil.la/724929.

[32] Nightingale, J. Mozilla Security Blog: DigiNotar Removal Follow Up. http://blog.mozilla.org/security/2011/09.

[33] Paxson, V. Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks 31*, 23-24 (1999).

[34] Prins, J. R. DigiNotar Certificate Authority Breach "Operation Black Tulip". Interim Report, Fox-IT, Sept. 2012.

[35] Sobey, J., Biddle, R., Oorschot, P. C., and Patrick, A. S. Exploring User Reactions to New Browser Cues for Extended Validation Certificates. In *ESORICS* (2008).

[36] Soghoian, C., and Stamm, S. Certified Lies: Detecting and Defeating Government Interception Attacks against SSL. *Financial Cryptography* (2012).

[37] Sotirov, A., Stevens, M., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D. A., and de Weger, B. MD5 Considered Harmful Today – Creating a Rogue CA Certificate, Dec. 2008. http://www.win.tue.nl/hashclash/rogue-ca/.

[38] Sotirov, A., and Zusman, M. Breaking the Myths of Extended Validation SSL Certificates. Black Hat USA Talk, 2009.

[39] Vratonjic, N., Freudiger, J., Bindschaedler, V., and Hubaux, J.-P. The Inconvenient Truth about Web Certificates. In *WEIS* (2011).

[40] Wendlandt, D., Andersen, D. G., and Perrig, A. Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing. In *USENIX Annual Technical Conference* (2008).

[41] Zusman, M. Criminal Charges are not pursued: Hacking PKI. DEFCON 17 Talk, 2009.