Analysis of the HTTPS Certificate Ecosystem[*]

Zakir Durumeric, James Kasten, Michael Bailey, J. Alex Halderman
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109, USA
{zakir, jdkasten, mibailey, jhalderm}@umich.edu

ABSTRACT

We report the results of a large-scale measurement study of the HTTPS certificate ecosystem—the public-key infrastructure that underlies nearly all secure web communications. Using data collected by performing 110 Internet-wide scans over 14 months, we gain detailed and temporally fine-grained visibility into this otherwise opaque area of security-critical infrastructure. We investigate the trust relationships among root authorities, intermediate authorities, and the leaf certificates used by web servers, ultimately identifying and classifying more than 1,800 entities that are able to issue certificates vouching for the identity of any website. We uncover practices that may put the security of the ecosystem at risk, and we identify frequent configuration problems that lead to user-facing errors and potential vulnerabilities. We conclude with lessons and recommendations to ensure the long-term health and security of the certificate ecosystem.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: [Network Protocols]; C.2.3 [**Computer-Communication Networks**]: [Network Operations]; E.3 [**Data Encryption**]: [Public key cryptosystems, Standards]

Keywords

TLS; SSL; HTTPS; public-key infrastructure; X.509; certificates; security; measurement; Internet-wide scanning

1. INTRODUCTION

Nearly all secure web communication takes place over HTTPS including online banking, e-mail, and e-commerce transactions. HTTPS is based on the TLS encrypted transport protocol and a supporting public key infrastructure (PKI) composed of thousands of certificate authorities (CAs)—entities that are trusted by users' browsers to vouch for the identity of web servers. CAs do this by signing digital certificates that associate a site's public key with its domain name. We place our full trust in each of these CAs—in general, every CA has the ability to sign trusted certificates for *any* domain, and so the entire PKI is only as secure as the weakest CA. Nevertheless, this complex distributed infrastructure is strikingly opaque. There is no published list of signed website certificates or even of the organizations that have trusted signing ability. In this work, we attempt to rectify this and shed light on the HTTPS certificate ecosystem.

Our study is founded on what is, to the best of our knowledge, the most comprehensive dataset of the HTTPS ecosystem to date. Between June 2012 and August 2013, we completed 110 exhaustive scans of the public IPv4 address space in which we performed TLS handshakes with all hosts publicly serving HTTPS on port 443. Over the course of 14 months, we completed upwards of 400 billion SYN probes and 2.55 billion TLS handshakes, collecting and parsing 42.4 million unique X.509 certificates from 109 million hosts. On average, each of our scans included 178% more TLS hosts and 115% more certificates than were collected in earlier studies of the certificate authority ecosystem [14], and we collected 736% more unique certificates in total than any prior study of HTTPS [16].

Using this dataset, we investigate two classes of important security questions, which relate to the behavior of CAs and to site certificates.

Certificate Authorities We analyze the organizations involved in the HTTPS ecosystem and identify 1,832 CA certificates, which are controlled by 683 organizations including religious institutions, museums, libraries, and more than 130 corporations and financial institutions. We find that more than 80% of the organizations with a signing certificate are not commercial certificate authorities and further investigate the paths through which organizations are acquiring signing certificates. We investigate the constraints on these CA certificates and find that only 7 CA certificates use name constraints, and more than 40% of CA certificates have no path length constraint. We identify two sets of misissued CA certificates and discuss their impact on the security of the ecosystem.

Site Certificates We analyze leaf certificates used by websites and find that the distribution among authorities is heavily skewed towards a handful of large authorities, with three organizations controlling 75% of all trusted certificates. Disturbingly, we find that the compromise of the private key used by one particular intermediate certificate would require 26% of HTTPS websites to immediately obtain new certificates. We provide an up-to-date analysis on the keys and signatures being used to sign leaf certificates and find that half of trusted leaf certificates contain an inadequately secure 1024-bit RSA key in their trust chain and that CAs were continuing

[*] Permission to make digital or hard copies of part or all of this work is granted without fee provided that copies are not distributed for profit and that copies bear this notice and the full citation on the first page. Copyright is held by the authors. This more permissive statement supersedes the ACM-mandated statement below, which has no effect. Please join us in supporting open access publication.

IMC'13, October 23–25, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-1953-9/13/10 ...$15.00.
http://dx.doi.org/10.1145/2504730.2504755.

to sign certificates using MD5 as late as April 2013. We find that 5% of trusted certificates are for locally scoped names or private IP address space (and therefore do not protect against man-in-the-middle attacks) and that 12.7% of hosts serving certificates signed by trusted CAs are serving them in a manner that will cause errors in one or more modern web browsers.

Lastly, we examine adoption trends in the HTTPS ecosystem from the past year, discuss anomalies we noticed during our analysis, and provide high-level lessons and potential paths forward to improve the security of the HTTPS ecosystem security. We ultimately hope that this global perspective and our analysis will inform future decisions within the security community as we work towards a more secure PKI. In order to facilitate future research on this critical ecosystem, we are releasing our dataset to the research community, including 42 million certificates and historical records of the state of 109 million HTTPS server IP addresses. This data and up-to-date metrics can be found at https://httpsecosystem.org/.

2. BACKGROUND

In this section, we present a brief review of TLS, digital certificates and their respective roles within the HTTPS ecosystem. We recommend RFC 5280 [11] for a more in-depth overview of the TLS public key infrastructure.

Transport Layer Security (TLS) Transport Layer Security (TLS) and its predecessor Secure Sockets Layer (SSL) are cryptographic protocols that operate below the application layer and provide end-to-end cryptographic security for a large number of popular application protocols, including HTTPS, IMAPS, SMTP, and XMPP [12]. In the case of HTTPS, when a client first connects, the client and server complete a TLS handshake during which the server presents an X.509 digital certificate, which is used to help identify and authenticate the server to the client. This certificate includes the identity of the server (e.g. website domain), a temporal validity period, a public key, and a digital signature provided by a trusted third party. The client checks that the certificate's identity matches the requested domain name, that the certificate is within its validity period, and that the digital signature of the certificate is valid. The certificate's public key is then used by the client to share a session secret with the server in order to establish an end-to-end cryptographic channel.

Certificate Authorities Certificate authorities (CAs) are trusted organizations that issue digital certificates. These organizations are responsible for validating the identity of the websites for which they provide a digital certificate. They cryptographically vouch for the identity of a website by digitally signing the website's *leaf certificate* using a browser-trusted *signing certificate*. Modern operating systems and web browsers ship with a set of these trusted *signing certificates*, which we refer to as *root certificates*. In all but a small handful of cases, all CAs are trusted unequivocally: a trusted CA can sign for *any* website. For example, a certificate for `google.com` signed by a German University is technically no more or less valid than a certificate signed by Google Inc., if both organizations control a trusted signing certificate.

The set of root authorities is publicly known because it is included with the web browser or operating system. However, root authorities frequently sign *intermediate certificates*, which generally retain all of the signing privileges of root certificates. This practice not only allows root authorities to store their signing keys offline during daily operation, but also allows authorities to delegate their signing ability to other organizations. When a server presents a leaf certificate, it must include the chain of authorities linking the leaf certificate to a trusted root certificate. This bundle of certificates is referred to as a *certificate chain*. We refer to certificates that have a valid chain back to a trusted root authority as *trusted certificates*. It is important to note that while intermediate authorities provide additional flexibility, the set of intermediate authorities is not publicly known until they are found in the wild—we ultimately do not know the identity of the organizations that can sign any browser-trusted certificate.

3. RELATED WORK

Several groups have previously studied HTTPS deployment and the certificate ecosystem. Most similar to our work, Holz et al. published a study in 2011 that focused on the dynamics of leaf certificates and the distribution of certificates among IP addresses, and attempted to roughly classify the overall quality of served certificates. The study was based on regular scans of the Alexa Top 1 Million Domains [1] and through passive monitoring of TLS traffic on the Munich Scientific Research Network [17]. The group collected an average 212,000 certificates per scan and a total 554,292 unique certificates between October 2009 and March 2011, approximately 1.3% of the number we have seen in the past year. Their passive experiments resulted in an average of 130,000 unique certificates. The aggregate size across both datasets was not specified.

We are aware of two groups that have performed scans of the IPv4 address space in order to analyze aspects of the certificate ecosystem. In 2010, the Electronic Frontier Foundation (EFF) and iSEC partners performed a scan over a three-month period as part of their SSL Observatory Project [14]. The project focused on identifying which organizations controlled a valid signing certificate. The EFF provided the first recent glimpse into the HTTPS certificate ecosystem, and while their study was never formally published, we owe the inspiration for our work to their fascinating dataset. Heninger et al. later performed a scan of the IPv4 address space in 2012 as part of a global study on cryptographic keys [16]. Similarly, Yilek et al. performed daily scans of 50,000 TLS servers over several months to track the Debian weak key bug [35]. We follow up on the results provided in these earlier works, adding another data point in the study of Debian weak keys and other poorly generated keys.

Most recently, Akhawe et al. published a study focusing on the usability of TLS warnings presented by web servers, deriving the logic used by web browsers to validate certificates, and making recommendations on how to better handle these error conditions [4]. Akhawe et al. also discuss differences in how OpenSSL and Mozilla NSS validate certificates, which we arrived at simultaneously.

Our study differs from previous work in the methodology we applied, the scope of our dataset, and the focus of our questions. While Holz et al. explored several similar questions on the dynamics of leaf certificates, the dataset we consider is more than 40 times larger, which we believe provides a more comprehensive view of the certificate ecosystem. The certificates found by scanning the Alexa Top 1 Million Domains provide one perspective on the CA ecosystem that is weighted towards frequently accessed websites. However, many of the questions we address are dependent on a more comprehensive viewpoint. The CA ecosystem is equally dependent on all certificate authorities and, as such, we are interested in not only the most popular sites (which are likely to be well configured) but also the potentially less visible certificates used by smaller sites and network devices. This difference is clearly visible in the number of CA certificates seen among the Alexa Top 1 Million sites. If our study had been founded only on these domains, we would have seen less than 30% of the trusted certificate authorities we uncovered, providing us with a less accurate perspective on the state of the ecosystem. Similarly, we build on many topics touched on by the EFF study, but we present updated and revised results, finding more than 3.5 times the number of hosts serving HTTPS than were seen three years ago and a changed ecosystem. Ultimately, we consider a different set of questions that are more focused on the dynamics of CAs and the certificates they sign, using a dataset that we believe provides a more complete picture than any previous study.

4. METHODOLOGY

Our data collection (which is ongoing as this paper goes to press) involves repeatedly surveying the certificate ecosystem through comprehensive scans of the IPv4 address space conducted at regular intervals. In this section, we describe how we perform these scans, collect and validate X.509 certificates, and finally, analyze our data.

Each scan consists of three stages: (1) discovering hosts with port 443 (HTTPS) open by enumerating the public address space, (2) completing a TLS handshake with responsive addresses and collecting the presented certificate chains, and (3) performing certificate parsing and validation. The scan process requires 18 hours to complete, including flushing all changes to the backend database, and is implemented in approximately 13,000 SLOC of C. The scans in this work were conducted using the regular office network at the University of Michigan Computer Science and Engineering division, from a single Dell Precision workstation with a quad-core Intel Xeon E5520 processor and 24 GB of memory. The access layer of the building runs at 10 Gbps and the building uplink to the rest of the campus is an aggregated 2 × 10 gigabit port channel.

4.1 Host Discovery

In the first stage of each scan, we find hosts that accept TCP connections on port 443 (HTTPS) by performing a single-packet TCP SYN scan of the public IPv4 address space using ZMap [13]. We choose to utilize ZMap based on its performance characteristics—ZMap is capable of completing a single packet scan of the IPv4 address space on a single port in approximately 45 minutes. Using ZMap, we send a single TCP SYN packet to every public IPv4 address and add hosts that respond with a valid SYN-ACK packet to an in-memory Redis queue for further processing. Our previous work finds an approximate 2% packet drop rate when performing single packet scans on our network [13]. In order to reduce the impact of packet loss on our long-term HTTPS results, we also consider hosts that successfully completed a TLS handshake in the last 30 days for follow-up along with the hosts found during the TCP SYN scan.

4.2 Collecting TLS Certificates

In the second processing stage, we complete a TLS handshake with the hosts we identified in the first stage and retrieve the presented certificate chain. We perform these TLS handshakes in an event-driven manner using libevent and OpenSSL [24,33]. Specifically, we utilize libevent's OpenSSL-based bufferevents, which allow us to define a callback that is invoked after a successful OpenSSL TLS negotiation. The retrieval process runs in parallel to the TCP SYN scan and maintains 2,500 concurrent TLS connections.

In order to emulate browser validation, we designed a custom validation process using the root browser stores from Apple Mac OS 10.8.2, Windows 7, and Mozilla Firefox. We find that a large number of web servers are misconfigured and present incomplete, misordered, or invalid certificate chains. OpenSSL validates certificates in a more stringent manner than most web browsers, including Mozilla Firefox and Google Chrome, which utilize Mozilla NSS [27] to perform certificate validation. To simulate the behavior of modern web browsers, we take the following corrective steps:

1. If the presented chain is invalid, we attempt to reorder the certificate chain. This resolves the situation when the correct intermediate certificates are provided, but are in the incorrect order.
2. We add previously seen intermediate authorities into OpenSSL's root store. This allows us to validate any certificate signed by a previously encountered intermediate CA regardless of the presented certificate chain.
3. Following each scan, we check certificates without a known issuer against the set of known authorities and revalidate any children for which there is a newly found issuer. This resolves the case where an intermediate is later found in a subsequent scan.

We parse collected TLS certificates using OpenSSL and maintain a PostgreSQL database of parsed data and historical host state.

4.3 Reducing Scan Impact

We recognize that our scans can inadvertently trigger intrusion detection systems and may upset some organizations. Many network administrators perceive port scans as the preliminary step in a targeted attack and in most cases are unable to recognize that their systems are not being uniquely targeted or that our research scans are not malicious in nature.

In order to minimize the impact of our scans and to avoid triggering intrusion detection systems, we scanned addresses according to a random permutation over a twelve hour period from a block of 64 sequential source IP addresses. When we perform a host discovery scan, an individual destination address receives at most one probe packet. At this scan rate, a /24-sized network receives a probe packet every 195 s, a /16 block every 0.76 s, and a /8 network block every 3 ms on average. In the certificate retrieval phase, we perform only one TLS handshake with each host that responded positively during host discovery.

In order to help users identify our intentions, we serve a simple webpage on all of the IP addresses we use for scanning that explains the purpose of our scanning and how to request that hosts be excluded from future scans. We also registered reverse DNS records that identify scanning hosts as being part of an academic research study. Throughout this study, we have coordinated with our local network administrators to promptly handle inquiries and complaints.

Over the course of 14 months, we received e-mail correspondence from 145 individuals and organizations. In most cases, notifications were informative in nature—primarily notifying us that we may have had infected machines—or were civil requests to be excluded from future scans. The vast majority of these requests were received at our institution's WHOIS abuse address or at the e-mail address published on the scanner IPs. In these cases, we responded with the purpose of our scans and excluded the sender's network from future scans upon request. Ultimately, we excluded networks belonging to 91 organizations or individuals and totaling 3,753,899 addresses (0.11% of the public IPv4 address space). Two requests originating from Internet service providers accounted for 49% of the excluded addresses. During our scans, we received 12 actively hostile responses that threatened to retaliate against our institution legally or via denial-of-service (DoS) attacks on network. In 2 cases we received retaliatory DoS traffic, which was automatically filtered by our upstream provider.

We discuss the ethical implications of performing active scanning and provide more details about the steps we take to reduce scan impact in our previous work [13].

4.4 Data Collection Results

We completed 110 successful scans of the IPv4 address space, completing 2.55 billion TLS handshakes, between June 6, 2012 and

Scan	EFF [14]	Ps & Qs [16]	First	Representative	Latest	Total
Date Completed	2010-8	2011-10	2012-6-10	2013-3-22	2013-8-4	Unique
Hosts with port 443 Open	16,200,000	28,923,800	31,847,635	33,078,971	36,033,088	(unknown)
Hosts serving HTTPS	7,704,837	12,828,613	18,978,040	21,427,059	24,442,824	108,801,503
Unique Certificates	4,021,766	5,758,254	7,770,385	8,387,200	9,031,798	42,382,241
Unique Trusted Certificates	1,455,391	1,956,267	2,948,397	3,230,359	3,341,637	6,931,223
Alexa Top 1 Mil. Certificates	(unknown)	89,953	116,061	141,231	143,149	261,250
Extd. Validation Certificates	33,916	71,066	89,190	103,170	104,167	186,159

Table 1: Internet-wide Scan Results — Between June 6, 2012 and August 4, 2013, we completed 110 scans of the IPv4 address space on port 443 and collected HTTPS certificates from responsive hosts.

August 4, 2013. Like to Holz et al. [17], we note that a large number of hosts on port 443 do not complete a TLS handshake. In our case we find that only 67% of hosts with port 443 open successfully complete a TLS handshake.

We retrieved an average of 8.1 million unique certificates during each scan, of which 3.2 million were browser trusted. The remaining 4.9 million untrusted certificates were a combination of self-signed certificates (48%), certificates signed by an unknown issuer (33%), and certificates signed by a known but untrusted issuer (19%). In total, we retrieved 42.4 million distinct certificates from 108.8 million unique IP addresses over the past eleven months. Of the hosts that performed complete TLS handshakes, an average of 48% presented browser-trusted X.509 certificates.

In our largest and most recent scan on August 4, 2013, we retrieved 9.0 million certificates from 24.4 million IP addresses of which 3.3 million were browser trusted. We show a comparison with previous work in Table 1. We also note that over 95% of trusted certificates and over 98% of hosts serving trusted certificates are located in only ten countries, shown in Table 2.

Country	Authorities	Certificates	Hosts
United States	30.34%	77.55%	75.63%
United Kingdom	3.27%	10.88%	18.15%
Belgium	2.67%	3.29%	1.51%
Israel	1.63%	2.56%	0.87%
Netherlands	2.18%	1.32%	0.49%
Japan	3.38%	1.06%	1.19%
Germany	21.28%	0.88%	0.35%
France	3.98%	0.38%	0.14%
Australia	0.81%	0.34%	0.11%
Korea	1.41%	0.24%	0.09%

Table 2: Top 10 Countries Serving Trusted Certificates

In this study, we choose to perform non-temporal analysis on the results from a representative scan, which took place on March 22, 2013 (highlighted column in Table 1). We choose to focus on the results from a single point-in-time instead of considering all certificates found over the past year due to varying lifespans. We find that organizations utilize certificates of differing validity periods and that in some cases, some devices have presented a different certificate in all of our scans. If we considered all certificates from the past year instead instead of what was hosted at a single point in time, these short lived certificates would impact the breakdown of several of our statistics.

4.5 Is Frequent Scanning Necessary?

Frequent repeated scans allow us to find additional certificates that would not otherwise be visible. We can illustrate this effect by considering the 36 scans we performed between January 1 and March 31, 2013 and analyzing the number of scans in which each certificate was seen. We find that 54% of browser-trusted certificates appeared in all 36 scans and that 70% of trusted certificates appear in more than 30 of our 36 scans. However, surprisingly, we find that 33% of self-signed certificates appeared in only one scan during the three month period. Many of these self-signed certificates appear to be served by embedded devices that generate new certificates on a regular basis. We found an average of 260,000 new certificates per scan during this period. The distribution is shown in Figure 1. Ultimately, we find that there are considerable advantages to scanning more frequently in obtaining a global perspective on the certificates valid at any single point in time, as well as the changing dynamics of the ecosystem over extended periods.

4.6 Server Name Indication Deployment

Both Holz [17] and Akhawe [4] cite Server Name Indication (SNI) as one of the reasons they choose to scan the Alexa Top 1 Million Domains and perform passive measurement instead of performing full IPv4 scans. Server Name Indication is a TLS extension that allows a client to specify the hostname it is attempting to connect to from the start of the TLS negotiation [9]. This allows a server to present multiple certificates on a single IP address and to ultimately host multiple HTTPS sites off of the same IP address that do not share a single certificate. Because we connect to hosts based on IP address in our scans and not by hostname, we would potentially miss any certificates that require a specific hostname.

In order to better understand the deployment of SNI and its impact on our results, we scanned the Alexa 1 Million Domains [1] using the same methodology we used for scanning the IPv4 address space. Of the Alexa Top 1 Million Domains, 323,502 successfully performed TLS handshakes and 129,695 of the domains presented browser-trusted certificates. Of the domains that completed a TLS handshake, only 0.7% presented certificates we had not previously seen in the most recent scan of the IPv4 address space. We cannot bound the number of hosts missed due to the deployment of SNI and it is clear that a small number of websites are adopting SNI, but we believe that our results are representative of certificate usage patterns. One reason SNI has not seen widespread deployment is because Internet Explorer on Windows XP does not support SNI. Although Windows XP market share is on the decline, it still represents more than a third of all operating system installations [26].

5. CERTIFICATE AUTHORITIES

The security of the HTTPS ecosystem is ultimately dependent on the set of CAs that are entrusted to sign browser-trusted certificates. Except in a small handful of cases, any organization with control of a signing certificate that chains to a browser-trusted root can sign a leaf certificate for any domain. As such, the entire ecosystem is

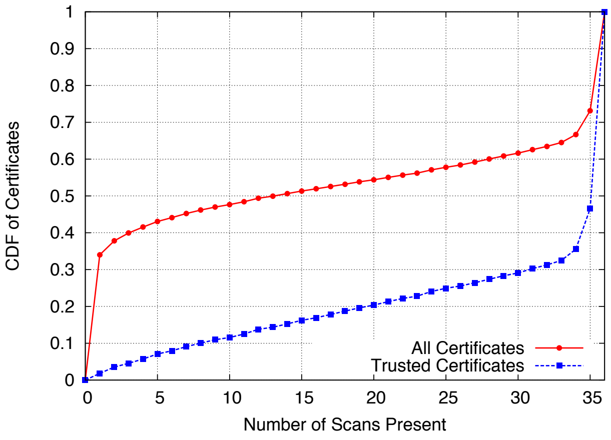

Figure 1: CDF of Scan Presence by Certificate — We performed 36 scans from 1/2013 to 3/2013. Here, we show the number of scans in which each certificate was found. We note that over 30% of self-signed certificates were only found in one scan.

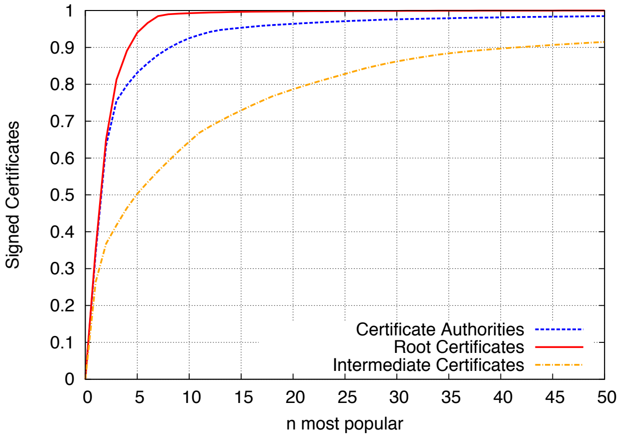

Figure 2: CDF of Leaf Certificates by CA — We find that 90% of trusted certificates are signed by 5 CAs, are descendants of 4 root certificates, and were signed by 40 intermediate certificates.

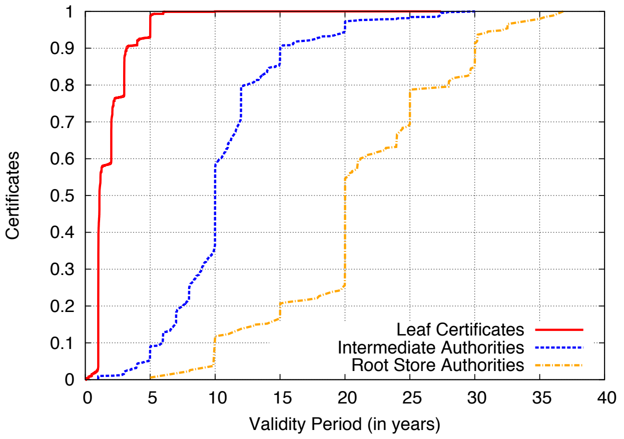

Figure 3: Validity Periods of Browser Trusted Certificates — Trusted CA certs are being issued with validity periods as long as 40 years, far beyond the predicted security of the keys they contain.

as fragile as the weakest CA. However, because there is no central, public registry of browser-trusted intermediate authorities, the organizations that control these signing certificates may be unknown until certificates they have signed are spotted in the wild. In this section, we describe the CAs we found during our scans and some of the practices they employ.

5.1 Identifying Trusted Authorities

We observed 3,788 browser-trusted signing certificates between April 2012 and August 2013 of which 1,832 were valid on March 22, 2013. All but seven of these signing certificates can sign a valid browser-trusted certificate for any domain. This is 25% more than were found by the EFF in 2010 and more than 327% more than were found by Ristic [31]. Holz et al. find 2,300 intermediate certificates in their active scanning [17]. However, this count appears to represent both browser-trusted and untrusted intermediates, of which we find 121,580 in our March 22 scan and 417,970 over the past year. While the raw number of signing certificates and HTTPS ecosystem as a whole have grown significantly over the past three years, we were encouraged to find that the number of identified organizations has not grown significantly.

These 1,832 signing certificates belong to 683 organizations and are located in 57 countries. While a large number of countries have jurisdiction over at least one trusted browser authority, 99% of the authorities are located in only 10 countries. We show the breakdown in Table 2. We classified the types of the organizations that control a CA certificate, which we show in Table 3. We were surprised to find that religious institutions, museums, libraries, and more than 130 corporations and financial institutions currently control an unrestricted CA certificate. Only 20% of organizations that control signing certificates are commercial CAs. We were unable to identify 15 signing certificates due to a lack of identification information or ambiguous naming. We also note that while there has been a 2% increase in the raw number of valid signing certificates over the past year, we have found negligible change in the number of organizations with control of a signing certificate.

5.2 Sources of Intermediates

Organizations other than commercial CAs control 1,350 of the 1,832 (74%) browser-trusted signing certificates, which raises the question of who is providing intermediate certificates to these organizations. We find that 276 of the 293 academic institutions along with all of the libraries, museums, healthcare providers, and religious institutions were signed by the German National Research and Education Network (DFN), which offers intermediate certificates to all members of the German network. DFN provided CA certificates to 311 organizations in total, close to half of the organizations we identified. While DFN has provided a large number of intermediate authorities to German institutions, we find no evidence that any are being used inappropriately. However, as we will discuss in Section 9, the attack surface of the certificate ecosystem could be greatly reduced by limiting the scope of these signing certificates.

The largest commercial provider of intermediate certificates is GTE CyberTrust Solutions, Inc., a subsidiary of Verizon Business, which has provided intermediate signing certificates to 49 third-party organizations ranging from Dell Inc. to Louisiana State University. Comodo (under the name *The USERTRUST Network*) provided intermediates to 42 organizations and GlobalSign to 20. We also saw a number of commercial authorities that provided a smaller number of certificates to seemingly unrelated entities. For example, VeriSign, Inc. provided intermediates for Oracle, Symantec, and the U.S. Government; SwissSign AG provided certificates for Nestle, Trend Micro, and other Swiss companies; StartCom Ltd.

Organization Type	Organizations		Authorities		Leaf Certificates		Hosts	
Academic Institution	273	(39.79%)	292	(15.93%)	85,277	(2.46%)	85,277	(0.92%)
Commercial CA	135	(19.67%)	819	(44.70%)	3,260,454	(94.20%)	3,260,454	(76.33%)
Government Agency	85	(12.39%)	250	(13.64%)	17,865	(0.51%)	17,865	(0.23%)
Corporation	83	(12.09%)	191	(10.42%)	30,115	(0.87%)	30,115	(4.80%)
ISP	30	(4.37%)	58	(3.16%)	8,126	(0.23%)	8,126	(1.55%)
IT/Security Consultant	29	(4.22%)	88	(4.80%)	22,568	(0.65%)	22,568	(0.98%)
Financial Institution	17	(2.47%)	49	(2.67%)	2,412	(0.06%)	2,412	(0.03%)
Unknown	*unknown*		15	(0.81%)	2,535	(0.07%)	2,535	(0.02%)
Hosting Provider	7	(1.02%)	12	(0.65%)	10,598	(0.30%)	10,598	(14.70%)
Nonprofit Org	7	(1.02%)	15	(0.81%)	11,480	(0.33%)	11,480	(0.11%)
Library	5	(0.72%)	6	(0.32%)	281	(0.00%)	281	(0.00%)
Museum	4	(0.58%)	4	(0.21%)	35	(0.00%)	35	(0.00%)
Healthcare Provider	3	(0.43%)	4	(0.21%)	173	(0.00%)	173	(0.00%)
Religious Institution	1	(0.14%)	1	(0.05%)	11	(0.00%)	11	(0.00%)
Military	1	(0.14%)	27	(1.47%)	9,017	(0.26%)	9,017	(0.27%)

Table 3: Types of Organizations with Signing Certificates — We found 1,832 valid browser-trusted signing certificates belonging to 683 organizations. We classified these organizations and find that more than 80% of the organizations that control a signing certificate are not commercial certificate authorities.

Parent Company	Signed Leaf Certificates	
Symantec	1,184,723	(34.23%)
GoDaddy.com	1,008,226	(29.13%)
Comodo	422,066	(12.19%)
GlobalSign	170, 006	(4.90%)
DigiCert Inc	145,232	(4.19%)
StartCom Ltd.	88,729	(2.56%)
Entrust, Inc.	76,990	(2.22%)
Network Solutions	62,667	(1.81%)
TERENA	42,310	(1.22%)
Verizon Business	32,127	(0.92%)

Table 4: Top Parent Companies — Major players such as Symantec, GoDaddy, and Comodo have acquired smaller CAs, leading to the 5 largest companies issuing 84.6% of all trusted certificates.

Organization	Signed Leaf Certificates	
GoDaddy.com, Inc.	913,416	(28.6%)
GeoTrust Inc.	586,376	(18.4%)
Comodo CA Limited	374,769	(11.8%)
VeriSign, Inc.	317,934	(10.0%)
Thawte, Inc.	228,779	(7.2%)
DigiCert Inc	145,232	(4.6%)
GlobalSign	117,685	(3.7%)
Starfield Technologies	94,794	(3.0%)
StartCom Ltd.	88,729	(2.8%)
Entrust, Inc.	76929	(2.4%)

Table 5: Top Certificate Authorities — The top 10 commercial certificate authorities control 92.4% of trusted certificates present in our March 22, 2013 scan.

provided certificates for The City of Osmio, Inc. and WoSign, Inc; QuoVadis Limited provided certificates for Migros and the Arab Bank Switzerland Ltd.; Entrust.net provided signing certificates to Disney, Experian PLC, and TDC Internet; and Equifax provided intermediates to Google Inc. This is not a clandestine practice, and several CAs advertise the sale of subordinate CA certificates.

Several corporations had a company authority in browser root stores. Approximately 30 of the 149 certificates in the Mozilla NSS root store belonged to institutions that we did not classify as commercial CAs, including Visa, Wells Fargo, Deutsche Telekom AG and the governments of France, Taiwan, Hong Kong, Japan, Spain, and the United States.

5.3 Distribution of Trust

While there are 683 organizations with the ability to sign browser-trusted certificates, the distribution is heavily skewed towards a small number of large commercial authorities in the United States. The security community has previously expressed concern over the sheer number of signing certificates [14], but it also worth considering the distribution of certificates among various authorities. An increasing number of signing certificates may in fact be a healthy sign if it indicates that authorities are using the new certificates in order to reduce the impact of compromise.

As shown in Figure 2, we find that more than 90% of browser-trusted certificates are signed by the 10 largest commercial CAs, are descendants of just 4 root certificates, and are directly signed by 40 intermediate certificates. Several large companies have acquired many of the smaller, previously independent commercial CAs. Symantec owns Equifax, GeoTrust, TC TrustCenter, Thawte, and VeriSign; GoDaddy owns Starfield Technologies and ValiCert; and Comodo owns AddTrust AB, eBiz Networks, Positive Software, RegisterFly, Registry Pro, The Code Project, The USERTRUST Network, WebSpace-Forum e.K., and Wotone Communications. These consolidations ultimately allow three organizations (Symantec, GoDaddy, and Comodo) to control 75% of the browser-trusted certificates seen in our study. We list the top 10 parent organizations in Table 4 and the top 10 commercial CAs in Table 5.

There is a long history of commercial CA compromise [8, 30, 32]. In each of these cases, web browsers and operating systems explicitly blacklisted the compromised signing certificate or misissued certificates [8, 28]. However, if a compromised signing certificate had signed for a substantial portion of the Internet, it would potentially be infeasible to revoke it without causing significant disruption to the HTTPS ecosystem [23]. As such, we would hope that large commercial authorities would distribute signing among a number of intermediate certificates. However, as seen in Figure 2, the exact opposite is true. More than 50% of all browser-trusted certificates have been directly signed by 5 intermediate certificates and a single intermediate certificate has signed 26% of currently valid HTTPS certs. If the private key for this intermediate authority were compromised, 26% of websites that rely on HTTPS would need to be immediately issued new certificates. Until these websites deployed the new certificates, browsers would present certificate warnings for all HTTPS communication. While it is not technically worrisome that a small number of organizations control a large percentage of the CA market, it is worrying that large CAs are not following simple precautions and are instead signing a large number of leaf certificates using a small number of intermediates.

5.4 Browser Root Certificate Stores

Microsoft, Apple, and Mozilla all maintain a distinct set of trusted signing certificates, which we refer to as root authorities. Google Chrome utilizes the OS root store in Windows and Mac OS and utilizes the root store maintained by Mozilla on Linux. Combined, the three groups trust 348 root authorities, but there are large discrepancies between the root certificates trusted by each organization. For example, as can be seen in Table 6, Windows trusts 125 additional authorities that are not present in any other OS or browser.

Systems Valid In	Roots	CAs	Signed
Windows Only	125	283	24,873
Mozilla Only	2	3	23
Apple Only	26	30	3,410
Windows & Mozilla	32	97	12,282
Windows & Apple	31	47	9,963
Mozilla & Apple	3	3	0
All Browsers	109	1,346	8,945,241

Table 6: Differences in Browser and OS Root Stores — While there are significant differences in the root certificates stores, 99.4% of trusted certificates are trusted in all major browsers.

The differences in the root stores lead to 463 partially trusted CAs. All but a small handful of the partially trusted authorities belong to government, regional, or specialty issuers. Only one of the partially trusted CAs, *ipsCA*, advertised itself as a commercial authority and sold certificates to the global market. Incidentally, the company claims to be "recognized by more than 98% of today's desktops" [2]. It fails to mention that its certificates are not trusted in Mozilla Firefox or on Mac OS.

Further investigation indicates that *ipsCA* was in the Mozilla root store in 2009, but was removed after several violations including the issuance of embedded-null prefix certificates, the unavailability of OCSP servers, and the issuance of leaf certificates with validity periods beyond the lifetime of the root CA certificate [34].

These 463 partially trusted authorities have little presence on the Internet. In total, they have signed certificates for only 51 domains in the Alexa Top 1 Million and for one domain in the Alexa Top 10,000 which belongs to *mci.ir*, an Iranian telecommunications company. Of the 348 root certificates, 121 of the authorities never signed any leaf certificates seen in our study, and 99.4% of the leaf certificates trusted by any browser are trusted in all browsers.

5.5 Name Constraints

While it is not an inherently poor idea to provide signing certificates to third-party organizations, these certificates should be restricted to a limited set of domains. Instead, all but 7 CAs in our March 22 scan can sign for any domain. X.509 Name Constraints [18] provide a technical mechanism by which parent authorities can limit the domains for which an intermediate signing certificate can sign leaf certificates. Optimally, signing certificates provided to third-party organizations, such as universities or corporations, would utilize name constraints to prevent potential abuse and to limit the potential damage if the signing certificate were compromised.

We find that only 7 trusted intermediate authorities out of 1,832 have name constraints defined, of which 3 were labeled as Comodo testing certificates. The remaining 4 are:

1. An intermediate provided by AddTrust AB to the Intel is limited to small a number of Intel owned domains.
2. An intermediate controlled by the U.S. State Department and provided by the U.S. Government root authority is prevented from signing certificates with the `.mil` top-level domain.
3. An intermediate provided to the Louisiana State University Health System is limited to a small number of affiliated domains.
4. A root certificate belonging to the Hellenic Academic and Research Institutions Certification Authority is restricted to the .gr, .eu, .edu, and .org domains.

5.6 Path Length Constraints

A signing authority can limit the number of intermediate authorities that can appear below it in a certificate chain by specifying an X.509 path length constraint [18] on the intermediate certificates that it signs. This is frequently used to prevent intermediate authorities from further delegating the ability to sign new certificates.

In our dataset, we find that 43% of signing certificates do not have any path length restriction defined. While this may not be a concern for large commercial CAs, we note that more than 80% of the intermediate authorities belonging to other types of organizations (e.g. corporations, academic, and financial institutions). While we saw little evidence of non-commercial CAs providing signing certificates to third-party organizations, we did observe governments using their intermediate authority to sign subordinate CA certificates for corporations within their country.

5.7 Authority Key Usage

All of the browser-trusted leaf certificates in our study were signed using an RSA key. As shown in Table 8, over 95% of browser trusted certificates were signed with 2048-bit RSA keys. We also note 6 browser-trusted authorities with ECDSA keys belonging to Symantec, Comodo, and Trend Micro. However, we found no trusted certificates that were signed using a ECDSA certificate.

Surprisingly, we find that 243 (13%) of the browser-trusted signing certificates were signed using a weaker key than they themselves contained. In all of these cases, the weakest key was the root authority. While only 58 (15.2%) of the 348 browser root authorities utilize 1024-bit RSA keys, these keys were used to indirectly sign 48.7% of browser-trusted certificates. In all of these cases, the CA organization also controlled a browser-trusted 2048-bit root certificate that could be used to re-sign the intermediate certificate.

NIST recommends that the public stop using 1024-bit keys in 2016 based on the expected computational power needed to compromise keys of this strength [5]. However, as seen in Figure 5, more than 70% of CA certificates using 1024-bit keys expire after this date and 57% of roots using 1024-bit RSA keys have signed children that expire after 2016. Figure 3 shows how certificate authorities are using certificates valid for up to 40 years—far beyond when their keys are expected to be compromisable. Most worryingly, it does not appear that CAs are moving from 1024-bit roots to more secure keys. As shown in Figure 4, we find only a 0.08% decrease in the number of certificates dependent on a 1024-bit root authority in the

Type	Root Authorities		Recursively Signed	
ECDSA	6	(1.8%)	0	(0%)
RSA (1024-bit)	53	(16.0%)	1,694,526	(48.6%)
RSA (2028-bit)	202	(61.0%)	1,686,814	(48.4%)
RSA (4096-bit)	70	(21.2%)	102,139	(2.9%)

Table 7: Key Distribution for Trusted Roots — The distribution of keys for root certificates shipped with major browsers and OSes.

Key Type	Authorities		Signed Leaves	
ECDSA	6	(0.3%)	0	(0%)
RSA (1024-bit)	134	(7.3%)	133,391	(4.2%)
RSA (2048-bit)	1,493	(78.9%)	3,034,751	(95.3%)
RSA (4096-bit)	198	(10.5%)	16,969	(0.5%)

Table 8: Key Distribution for Trusted Signing Certificates

past year. In 2012, 1.4 million new certificates were issued that were rooted in a 1024-bit authority, and 370,130 were issued between January and April 2013.

6. LEAF CERTIFICATES AND HOSTING

Over the last 14 months, we collected 6.93 million unique trusted certificates. In our March 22 scan, we observed 3.2 million unique trusted certificates from 21.4 million hosts. In this section, we discuss the dynamics of these trusted leaf certificates and the hosts serving them.

6.1 Keys and Signatures

Public Keys In line with previous studies, we find that over 99% of trusted leaf certificates contain RSA public keys. We provide a breakdown of leaf key types in Table 9. Over the course of the past year, we found 47 certificates that contain ECDSA public keys; none were present in our March 22 scan and none were browser trusted. Recently, Google began to use ECDSA certificates for several services. However, these sites are only accessible through the use of server name indication (SNI) and so do not appear in our dataset.

We find 2,631 browser-trusted certificates using 512-bit RSA keys, which are known to be easily factorable, and 73 certificates utilizing 768-bit keys, which have been shown to be factorable with large distributed computing efforts [20]. While a large number of these certificates were found being actively hosted, only 16 have not yet expired or been revoked. No browser-trusted authorities have signed any 512-bit RSA keys since August 27, 2012. We were further encouraged to find that less than 4% of valid trusted certificates used 1024-bit keys.

Weak Keys Previous studies have exposed the use of weak keys in the HTTPS space [16,22,35]. We revisit several of these measurements and provide up-to-date metrics. Following up on the study performed by Heninger et al. [16], we find that 55,451 certificates contained factorable RSA keys and are served on 63,293 hosts, a 40% decrease in the total percentage of hosts with factorable keys, but only a slight decrease (1.25%) in the raw number of hosts found using factorable keys since 2011. Three of the factorable certificates are browser trusted; the last was signed on August 9, 2012. 2,743 certificates contained a Debian weak key [7], of which 96 were browser trusted, a 34% decrease from 2011 [16]. The last browser-trusted certificate containing a Debian weak key was signed on January 25, 2012.

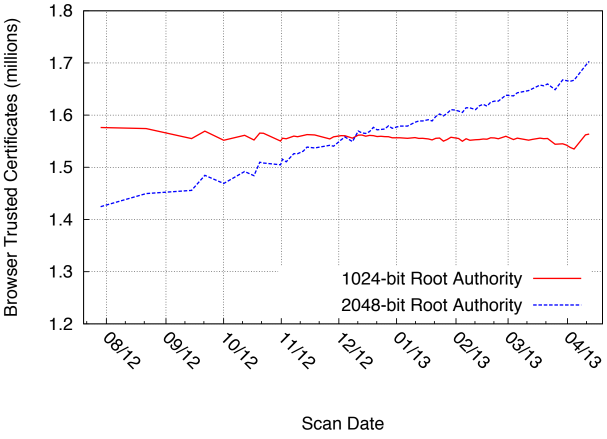

**Figure 4: Temporal Trends in Root Key Size — We find that 48.7% of browser-trusted leaf certificates are dependent on 1024-bit RSA based root authorities, contrary to recommended practice [5].

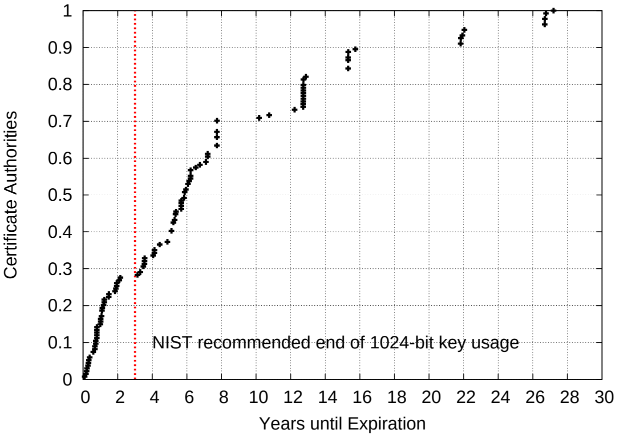

**Figure 5: Expiration of 1024-bit Root Certificates — This figure shows when trusted 1024-bit RSA CA certificates expire. We note that more than 70% expire after 2016 when NIST recommends discontinuing the use of 1024-bit keys.

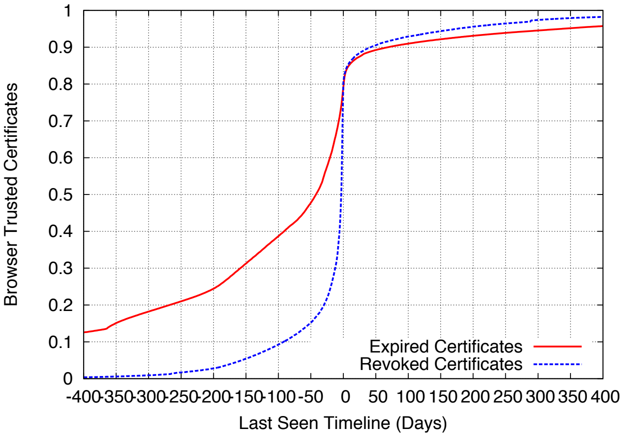

**Figure 6: CDF of Certificate Removal — We find that 20% of expiring certificates and 19.5% of revoked certificates are removed retroactively (to the right of 0 days).

Key Type	All Trusted		Valid Trusted
RSA (≤ 512-bit)	2,631	(0.1%)	16
RSA (768-bit)	73	(0.0%)	0
RSA (1024-bit)	341,091	(10.5%)	165,637
RSA (1032–2040-bit)	23,888	(0.7%)	105
RSA (2048-bit)	2,816,757	(86.4%)	2,545,693
RSA (2056–4088-bit)	1,006	(0.0%)	921
RSA (4096-bit)	74,014	(2.3%)	65,780
RSA (> 4096-bit)	234	(0.0%)	192
DSA (all)	17	(0.0%)	7
ECDSA (all)	0	(0.0%)	0

Table 9: Trusted Leaf Certificate Public Key Distribution

Type	Trusted Certificates	
SHA-1 with RSA Encryption	5,972,001	(98.7%)
MD5 with RSA Encryption	32,905	(0.54%)
SHA-256 with RSA Encryption	15,297	(0.25%)
SHA-512 with RSA Encryption	7	(0.00%)
MD2 with RSA Encryption	21	(0.00%)
Other	29,705	(0.49%)

Table 10: Trusted Leaf Certificate Signature Algorithms

Signature Algorithms In line with the results presented by Holz et al. [17], we find that 98.7% of browser-trusted certificates are signed using SHA-1 and RSA encryption. We find 22 trusted certificates with MD2-based signatures and 31,325 with MD5 signatures. Due to known weaknesses in these hash functions, no organizations should currently be using them to sign certificates. The last certificate signed with MD5 was issued on April 17, 2013 by Finmeccanica S.p.A., an Italian defense contractor, more than 4 years after Sotirov et al. published "MD5 considered harmful today" [32]. We provide a breakdown of leaf certificate signature types in Table 10.

Certificate Depth Similarly to the EFF and Holz et al., we find that the vast majority (98%) of leaf certificates are signed by intermediate authorities one intermediate away from a root authority. However we find that 61 root authorities directly signed 41,000 leaf certificates and that there exist leaf certificates as many as 5 intermediates away from a root authority. All but a handful of the authorities 4 or more intermediates away from a browser-trusted root belonged to agencies within the U.S. Federal Government.

We are not aware of any vulnerabilities created by having a long certificate chain. However, it is worrisome to see leaf certificates directly signed by root authorities, because this indicates that the root signing key is being actively used and may be stored in a network-attached system, raising the risk of compromise. If the signing key were to be compromised, the root certificate could not be replaced without updating all deployed browser installations. If an intermediate authority were used instead to sign these leaf certificates, then it could be replaced by the root authority without requiring browser updates, and the root could be kept offline during day-to-day operation.

6.2 Incorrectly Hosted Trusted Certificates

We find that 1.32 million hosts (12.7%) serving once-valid browser-trusted leaf certificates are misconfigured in a manner such that they are inaccessible to some clients or are being hosted beyond their validity period. We show a breakdown of reasons that certificates are invalid in Table 11. We note that Mozilla Network Security Services (NSS) [27], the certificate validation library utilized by many browsers, caches previously seen intermediates. Because of this, many certificates with invalid trust chains will appear valid in users' browsers if the intermediate authorities have previously been encountered.

Approximately 5.8% of hosts are serving now-expired certificates, which will be considered invalid by all browsers. We find that 22% of certificates are removed retroactively after their expiration and that 19.5% of revoked certificates are removed after they appear in a certificate revocation list (CRL). We show the distribution of when certificates are removed from servers in Figure 6. Another 42.2% of hosts are providing unnecessary certificates in the presented trust chain. Although this practice has no security implications, these additional certificates provide no benefit to the client and ultimately result in a slight performance degradation.

Holz et al. report that 18% of all certificates are expired. However, this statistic reflected all certificates, over 25% of which are self-signed and would already raise a browser error. We instead consider only certificates signed by browser-trusted authorities, which would otherwise be considered valid.

Status	Hosts	
Expired	595,168	(5.80%)
Not Yet Valid	1,966	(0.02%)
Revoked	28,033	(0.27%)
No Trust Chain	654,667	(6.30%)
Misordered Chain	25,667	(0.24%)
Incorrect Chain	11,761	(0.14%)
Unnecessary Root	4,365,321	(42.2%)
Optimally Configured	4,657,133	(45.0%)

Table 11: Common Server Certificate Problems — We evaluate hosts serving browser-trusted certificates and classify common certificate and server configuration errors. The number of misconfigured hosts indicates that procuring certificates and correctly configuring them on servers remains a challenge for many users.

6.3 Invalid Authority Types

We find that 47 (2.6%) of the 1,832 browser-trusted signing certificates are not denoted for signing TLS certificates for use on the web. Of these 47 signing certificates, 28 (60%) are designated for signing Microsoft or Netscape Server Gated Crypto certificates, a now obsolete cryptographic standard that was used in the 1990s in response to U.S. regulation on the export of strong cryptographic standards [29].

The remaining 19 signing certificates are designated for combinations of *Code Signing*, *E-mail Protection*, *TLS Web Client Authentication*, *Time Stamping*, and *Microsoft Encrypted File System*. These intermediate certificates were not found in any browser or operating system root stores but were found being served on public web servers. It does not appear that any of these authorities were signing certificates inappropriately; nobody was attempting to sign a TLS Web Server Authentication certificate using an authority marked for another use. Instead, we found that individuals and organizations were mistakenly using valid code signing and e-mail certificates as the TLS leaf certificate on their websites.

6.4 Certificate Revocation

Certificate authorities can denote that previously issued certificates should no longer be trusted by publishing their revocation in a public *certificate revocation list* (CRL). The location of authority CRLs are listed in each signed certificate. In order to understand why certificates are being revoked, we fetched and parsed the CRLs listed in all browser-trusted certificates. We find that 2.5% of browser-trusted certificates are eventually revoked by their authority. We present a breakdown of revocation reasons in Table 9. While RFC 5280 [11] strongly encourages issuers to provide "meaningful" reason codes for CRL entries, we find that 71.7% of issuers who revoked certificates do not provide reasons for any of their revocations.

While 2.5% of certificates are eventually revoked, we find that only 0.3% of hosts presenting certificates in our scan were revoked. We expect that this is because the site operators will request a certificate be revoked and simultaneously remove the certificate from the web server. As can be seen in Figure 6, more than 80% of certificates are removed proactively and were not seen again after the time of their revocation.

WebTrust for Certificate Authorities [3], an audit mandated by the three major root stores, requires that authorities maintain an online repository that allows clients to check for certificate revocation information. However, we find that 14 trusted signing certificates from 9 organizations fail to include revocation data in at least some of their certificates, and in 5 cases do not supply revocation data in any of their signed certificates.

7. UNEXPECTED OBSERVATIONS

We observed a variety of unexpected phenomenon during our scans over the past year. We describe these observations here.

7.1 CA Certs with Multiple Parents

Of the 1,832 browser-trusted signing certificates we found, 380 shared their subject, public key, and subject key identifier with another browser-trusted certificate forming 136 groups of "sibling" CA certificates. Because of this, leaf certificates can have more than one parent from the browsers' point of view. We find that only 37.4% of browser trusted leaf certificates have a single parent; 38.7% have two parents; 12.3% have three; 11.3% have four; and a small number have 5–9 valid parents. Depending on which parent is presented in a trust chain, the perceived validity of the leaf certificate can change. For example, if the presented intermediate certificate has expired, then the leaf certificate will be considered invalid. We note that subject key identified sometimes also specifies additional constraints such as a constraint on issuer serial number. However, we find that only a handful of certificates contain additional constraints.

In 86 of the 136 groups of sibling certificates, the signing certificates had differing validity periods. In four sets, one of the certificates was revoked, in a separate four sets, each authority was in a different browser or OS root store, and in 49 cases the authorities were signed by different parent authorities. While previous studies found evidence of this phenomenon, we were not aware of the prevalence of this behavior. We are not aware of any security vulnerabilities that are introduced by this practice, but we do find that 43,674 (1.35%) of the browser-trusted certificates are presented with the incorrect parent, which limits their perceived validity (e.g. the presented CA certificate expires earlier the leaf certificate, but another parent exists with a later expiration date).

7.2 CA Certs with Negative Path Lengths

We find that 1,395 browser-trusted CA certificates have a negative path length constraint, which renders them unable to sign any certificates due to a path length restriction earlier in the trust chain. These malformed intermediate certificates were signed by the Government of Korea and provided to educational institutions ranging

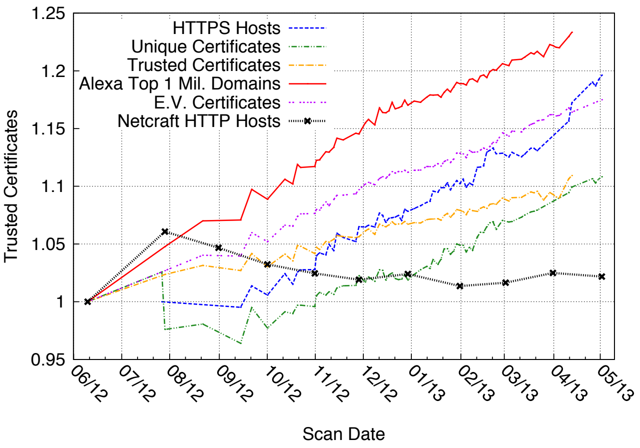

Figure 7: Growth in HTTPS Usage — Over the past 14 months, we observe between 10-25% growth of all aspects of HTTPS usage.

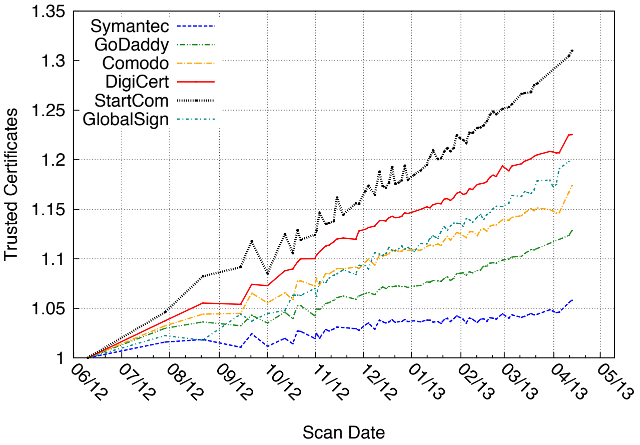

Figure 8: Change in Authority Market Share — In this figure, we should the individual growth of the top 10 most prolific certificate authorities.

Revocation Reason	Revoked Certificates	
Cessation Of Operation	101,370	(64.9%)
Not Provided	31514	(20.2%)
Affiliation Changed	7,384	(4.7%)
Privilege Withdrawn	5,525	(3.5%)
Unspecified	4,523	(2.9%)
Superseded	3,887	(2.5%)
Key Compromise	1,945	(1.2%)
Certificate Hold	45	(0.0%)
CA Compromise	2	(0.0%)
Total	**156,195**	

Figure 9: Reasons for Revocation — We find that 10,220 (2.5%) of the browser trusted certificates seen in our study were eventually revoked. Both of the "CA Compromised" revocations were due to the DigiNotar compromise [8].

from elementary schools to universities, libraries, and museums. However, because they are still technically CA certificates, web browsers including Mozilla Firefox and Google Chrome will not recognize them as valid leaf certificates.

We do not include these certificates when referring to the set of browser-trusted authorities because they are unable to sign any certificates and therefore do not have the same influence as other valid authorities. However, we note that some less common client implementations may fail to properly check the path length constraint and incorrectly treat these as valid. One of these CA certificates, issued to a Korean elementary school, was compromised by Heninger [15], who factored the 512-bit key a few hours after the certificate expired.

7.3 Mis-issued CA Certificates

We found one mis-issued signing certificate during the course of our study, which was issued for `*.EGO.GOV.TR`, by Turktrust, a small Turkish certificate authority. We found the certificate served as a leaf certificate on what appeared to be an unconfigured IIS server on a Turkish IP address. We saw 487 certificates that were signed by Turktrust over the course of our study. All were for Turkish organizations or the Turkish Government; we saw no evidence of other mis-issued certificates.

The certificate was later found by Google after being used to sign a Google wildcard certificate [21] and was revoked by Turktrust on December 26, 2012. It was last seen in our scans on December 27, 2012.

7.4 Site Certificates with Invalid Domains

We find that 4.6% (149,902) of browser-trusted certificates contain a common name (CN) or subject alternate name for a locally scoped domain or private IP address. Because these names are not fully qualified, the intended resource is ambiguous and there is no identifiable owner. As such, these local domain names frequently appear on more than one certificate. In one example, there are 1,218 browser-trusted certificates for the domain `mail` owned by organizations ranging from the U.S. Department of Defense to the Lagunitas Brewing Company.

The vast majority of certificates appear to be related to mail services. Of the 157,861 certificates with locally scoped names, 25,964 contain the name `exchange` (Microsoft Exchange Mail Server) and 99,773 contain a variation on the name `mail`. More than 100,000 of the certificates contain a domain ending in `.local`.

We suspect that certificates include these locally scoped names in order to facilitate users that are part of an Active Directory domain in connecting to their local Exchange mail server. In this scenario, the integrated DNS service in Active Directory will automatically resolve locally scoped names to the correct server on the domain. However, these clients will receive a name mismatch error if the TLS certificate presented by the Exchange Server does not match the locally scoped name that was originally resolved. Instead of requiring users to use the fully qualified domain name (FQDN) of the Exchange Server unlike other servers on the domain, certificate authorities include the local name of the Exchange server. In the case of certificates ending in `.local`, Active Directory Forests are generally rooted in an FQDN. In cases where organizations have not registered an FQDN for their forest, Active Directory elects to use the `.local` TLD.

Unfortunately, this practice does not provide security against man-in-the-middle attacks. It is trivial to procure a certificate with the same locally scoped name as another organization. Because there is no identifiable owner for the domain, both certificates are equally valid, and the subsequent certificate can be used to impersonate the original organization.

8. ADOPTION TRENDS

We observe a steady, linear increase in nearly all aspects of HTTPS adoption between June 2012 and April 2013, as shown in Figure 7. Most notably, there is a 23.0% increase in the number of Alexa Top 1 Million domains serving trusted certificates and a 10.9% increase in the number of unique browser-trusted certificates found during each scan. During this time, the Netcraft Web survey finds only a 2.2% increase in the number of active sites that respond over HTTP [25]. Based on the Netcraft Survey, we find an 8.5% increase in the number of websites utilizing HTTPS from 1.61% to 1.75%. This indicates that the increase in the number of certificates is not solely dependent on the growth of the Internet, but that there is an increase in the adoption of HTTPS in existing sites. We also note a 16.8% increase in the number of extended validation certificates, a 19.6% increase in the number of hosts serving HTTPS on port 443, and an 11.1% increase in the total number of TLS certificates over this period.

The market share of each authority did not change drastically over the past year. In terms of number of valid signed leaf certifictes, Symantec grew 6%, GoDaddy 13%, and Comodo 17%. During this time, there was a 10.9% increase in the global number of unique valid browser-trusted certificates. StartCom, a smaller authority based in Israel that offers free basic certificates, grew by 32% over the course of the year, from 2.17% to 2.56% market share. We plot the growth of the top authorities in Figure 8.

9. DISCUSSION

Analyzing the certificate authority ecosystem from a global perspective reveals several current practices that put the entire HTTPS ecosystem at risk. In this section, we discuss our observations and possible paths forward.

Ignoring Foundational Principles The security community has several widely accepted best practices such as the *principle of least privilege* and *defense in depth*. However, these guidelines are not being well applied within one of our most security critical ecosystems. For instance, there are several technical practices already at our disposal for limiting the scope of a signing certificate, including setting name or path length constraints and distributing leaf certificates among a large number intermediate certificates. There are clear cases for using these restrictions, but the vast majority of the time, CAs are not fully utilizing these options.

One example of how defense in depth successfully prevented compromise can be seen in the 1,400 signing certificates that were mis-issued to organizations in South Korea (Section 7.2). In this case, a path length constraint on a grandparent certificate prevented this error from becoming a massive vulnerability. To put this in context, if *defense in depth* had not been practiced, the erroneous action of a single certificate authority would have tripled the number of organizations controlling a valid signing certificate overnight. Unfortunately, while a path length constraint was in place for this particular situation, more than 40% of CA certificates do not have any constraints in place to prevent this type of error and only a small handful use name constraints.

In a less fortunate example, Turktrust accidentally issued a signing certificate to one of its customers that ultimately signed a valid certificate for `*.google.com` (Section 7.3). If name or path constraints had been applied to Turktrust's CA intermediate certificate, the incident could have been avoided or, at the very least, reduced in scope. In other situations, the risk associated with compromise of a single signing certificate could be decreased by simply spreading issuance across multiple certificates (Section 5.3).

Standards and Working Groups The CA/Browser Forum is a voluntary working group composed of certificate authorities and Internet browser software vendors. The group has recently attempted to resolve many of the security risks previously introduced by certificate authorities, and in November 2011, the it adopted guidelines for certificate authorities [10] that touch on many of the concerns we raise.

However with only 20% of the organizations controlling signing certificates being commercial certificate authorities and less than 25% of commercial authorities participating in the workgroup, there remains a disconnect. It is unclear how many organizations are aware of the existence of the baseline standard, but it is clear that a large number of organizations are either unaware or are choosing to ignore the forum's baseline requirements. One example of this non-adherence can be seen in the agreement to cease the issuance of certificates containing internal server names and reserved IP addresses. Despite the ratification of this policy, more than 500 certificates containing internal server names and which expire after November 1, 2015 have been issued since July, 1, 2012 by CA/B Forum members (Section 7.4).

Without any enforcement, members of the CA/Browser Forum have disregarded adopted policies and we expect that other organizations are unaware of the standards. There is still work required from the security community to reign in these additional authorities and to follow up with members that are disregarding existing policies.

Browsers to Lead the Way Web browser and operating system maintainers are in a unique position to set expectations for certificate authorities, and it is encouraging to see increasing dialogue in the CA/Browser Forum. However, browsers also have a responsibility to commit resources towards a healthier ecosystem. Many new, more secure technologies are dependent on support in common browsers and web servers. Without browser compatibility, certificate authorities lack incentive to adopt new, more secure options regardless of support from the security community.

This can immediately be seen in the deployment of name constraints. We find that the vast majority of the CA certificates issued to non-CAs are used to issue certificates to a small number of domains and, as such, could appropriately be scoped using name constraints with little impact on day-to-day operations. Restricted scopes have been shown to greatly reduce the attack surface of the CA ecosystem [19], and with 80% of existing signing certificates belonging to organizations other than commercial certificate authorities, there is a clear and present need for name constraints (Section 5). However, Safari and Google Chrome on Mac OS do not currently support the critical server name constraint extension. As a result, any certificate signed using an appropriately scoped CA certificate with the extension marked as critical will be rejected on these platforms. Therefore, while there is community consensus on the value of server name constraints, progress will be slow until all browsers support the extension.

Failing to Recognize Cryptographic Reality It is encouraging to find that over 95% of trusted leaf certificates and 95% of trusted signing certificates use NIST recommended key sizes [6]. However, more than 50 root authorities continue to use 1024-bit RSA keys, the last of which expires in 2040—more than 20 years past recommended use for a key of this size (Section 5.7). Authorities are not adequately considering long-term consequences of authority certificates and need to anticipate what the cryptographic landscape will be in the future. Many of these root certificates were signed prior to guidelines against such long-lived CA certificates. However, today, we need to be working to resolve these past errors and preparing to remove now-inappropriate root CAs from browser root stores.

10. CONCLUSION

In this work, we completed the largest known measurement study of the HTTPS certificate ecosystem by performing 110 comprehensive scans of the IPv4 HTTPS ecosystem over a 14 month period. We investigated the organizations that the HTTPS ecosystem depends on and identified several specific practices employed by certificate authorities that lead to a weakened public key infrastructure. We provided updated metrics on many aspects of HTTPS and certificate deployment along with adoption trends over the last year. Lastly, we discussed the high-level implications of our results and make several recommendations for strengthening the ecosystem. Our study shows that regular active scans provide detailed and temporally fine-grained visibility into this otherwise opaque area of security critical infrastructure. We are publishing the data from our scans at https://httpsecosystem.org/ in the hope that it will assist other researcher in further investigating the HTTPS ecosystem.

Acknowledgments

The authors thank the exceptional sysadmins at the University of Michigan for their help and support throughout this project. This research would not have been possible without Kevin Cheek, Chris Brenner, Laura Fink, Paul Howell, Don Winsor, and others from ITS, CAEN, and DCO. We thank the anonymous reviewers and our shepherd, Udi Weinsberg, for their insightful suggestions and comments. We also thank Brad Campbell, Peter Eckersley, Ralph Holz, Ben Laurie, Pat Pannuto, and Eric Wustrow. This work was supported in part by the Department of Homeland Security Science and Technology Directorate under contracts D08PC75388, FA8750-12-2-0314, and FA8750-12-2-0235; the National Science Foundation (NSF) under contracts CNS 1111699, CNS 091639, CNS 08311174, CNS 0751116, CNS 1330142, and CNS 1255153; and the Department of the Navy under contract N000.14-09-1-1042.

11. REFERENCES

[1] Alexa Top 1,000,000 Sites. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.

[2] Facts about ipsCA, April 2013. http://certs.ipsca.com/companyIPSipsCA/competitorssay.asp.

[3] WebTrust for Certification Authorities — SSL Baseline Requirements Audit Criteria v.1.1, Jan. 2013. http://www.webtrust.org/homepage-documents/item72056.pdf.

[4] 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 the World Wide Web*, pages 59–70, 2013.

[5] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for Key Management - Part 1: General (Revision 3), 2012. http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf.

[6] E. Barker and A. Roginsky. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, Jan. 2011. http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf.

[7] L. Bello. DSA-1571-1 OpenSSL—Predictable random number generator, 2008. Debian Security Advisory. http://www.debian.org/security/2008/dsa-1571.

[8] S. Bhat. Gmail users in Iran hit by MITM Attacks, Aug. 2011. http://techie-buzz.com/tech-news/gmail-iran-hit-mitm.html.

[9] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, and T. Wright. Transport Layer Security (TLS) Extensions. RFC 3546 (Proposed Standard), June 2003.

[10] CA/Browser Forum. Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, v.1.1, Nov. 2011. https://www.cabforum.org/Baseline_Requirements_V1_1.pdf.

[11] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard), May 2008. Updated by RFC 6818.

[12] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008. Updated by RFCs 5746, 5878, 6176.

[13] 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.

[14] P. Eckersley and J. Burns. An observatory for the SSLiverse. Talk at Defcon 18 (2010). https://www.eff.org/files/DefconSSLiverse.pdf.

[15] N. Heninger. Factoring as a service, Aug. 2013. Talk at CRYPTO Rump Session 2013. http://crypto.2013.rump.cr.yp.to/981774ce07e51813fd4466612a78601b.pdf.

[16] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *21st USENIX Security Symposium*, Aug. 2012.

[17] 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 *11th ACM SIGCOMM conference on Internet measurement (IMC)*, 2011.

[18] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 public key infrastructure certificate and CRL profile. (2459), Jan. 2009.

[19] J. Kasten, E. Wustrow, and J. A. Halderman. Cage: Taming certificate authorities by inferring restricted scopes. In *17th International Conference on Financial Cryptography and Data Security (FC)*, 2013.

[20] T. Kleinjung, K. Aoki, J. Franke, A. Lenstra, E. Thomé, J. Bos, P. Gaudry, A. Kruppa, P. Montgomery, D. Osvik, et al. Factorization of a 768-bit rsa modulus. In *Advances in Cryptology–CRYPTO 2010*, pages 333–350. Springer, 2010.

[21] A. Langley. Enhancing digital certificate security. Google Online Security Blog, http://googleonlinesecurity.blogspot.com/2013/01/enhancing-digital-certificate-security.html, Jan. 2013.

[22] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter. Ron was wrong, Whit is right. *IACR Cryptology ePrint Archive*, 2012:64, 2012.

[23] M. Marlinspike. SSL and the future of authenticity, Aug. 2011. Talk at BlackHat 2011. http://www.thoughtcrime.org/blog/ssl-and-the-future-of-authenticity/.

[24] N. Mathewson and N. Provos. libevent—An event notification library. http://libevent.org.

[25] Netcraft, Ltd. Web server survey. http://news.netcraft.com/archives/2013/05/03/may-2013-web-server-survey.html, May 2013.

[26] NetMarketShare. Desktop operating system market share, Apr. 2013. http://www.netmarketshare.com/operating-system-market-share.aspx.

[27] M. D. Network. Mozilla network security services (nss). http://www.mozilla.org/projects/security/pki/nss/.

[28] J. Nightingale. Comodo Certificate Issue – Follow Up, Mar. 2011. https://blog.mozilla.org/security/2011/03/25/comodo-certificate-issue-follow-up/.

[29] E. Rescorla. *SSL and TLS: designing and building secure systems*, volume 1. Addison-Wesley Reading, 2001.

[30] R. Richmond. Comodo fraud incident, Mar. 2011. http://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html.

[31] I. Ristic. Internet SSL survey 2010. Talk at BlackHat 2010. http://media.blackhat.com/bh-ad-10/Ristic/BlackHat-AD-2010-Ristic-Qualys-SSL-Survey-HTTP-Rating-Guide-slides.pdf.

[32] A. Sotirov, M. Stevens, J. Appelbaum, A. Lenstra, D. Molnar, D. A. Osvik, and B. de Weger. MD5 considered harmful today. http://www.win.tue.nl/hashclash/rogue-ca/.

[33] J. Viega, M. Messier, and P. Chandra. *Network Security with OpenSSL: Cryptography for Secure Communications*. O'Reilly, 2002.

[34] K. Wilson. Bug 523652 - IPS action items re IPS SERVIDORES root certificate, Nov. 2009. https://bugzilla.mozilla.org/show_bug.cgi?id=523652.

[35] S. Yilek, E. Rescorla, H. Shacham, B. Enright, and S. Savage. When private keys are public: results from the 2008 Debian OpenSSL vulnerability. In *2009 ACM SIGCOMM Internet Measurement Conference*, pages 15–27.