Reigning in the Wild West of TLS Certificate Provisioning

GIAC GCCC Gold Certification

Author: Blaine Hein, blaine.hein@gmail.com
Advisor: Bryan Simon

Accepted: July 24, 2019

Abstract

Several Internet Engineering Task Force (IETF) protocols and data structures claim to solve the security issues surrounding the use of Transport Layer Security (TLS) and Secure Sockets Layer (SSL) on the web. DNS-based Authentication of Named Entities (DANE) implements a new TLSA resource record which names the expected certificate a browser should receive for a given TLS server. DNS Certification Authority Authorization (CAA) resource records constrain the Certificate Authority permission to issue a certificate for a specific TLS server. Certificate transparency leverages publication of certificates in public log servers to show rogue TLS certificates. Some of these proposals have failed to receive broad support across vendors, while others may not adequately address the vulnerabilities they are meant to fix. Navigating through these protocols in search of a set of solutions for TLS security within an organization is not trivial. This paper analyses the evolution of TLS security services and shows the remaining threats against them.

1. Introduction

Managing TLS services within an enterprise has never been a simple task. The number of TLS enabled computers have grown exponentially in the last decade (2009-2019), and the growth of outsourced networks and services has moved these endpoints outside of the traditional data center. Large scale exploits against TLS using vulnerabilities in DNS, Certificate Authority (CA) security, and CA business processes occurred between 2011 and 2015. While TLS security mechanisms have matured dramatically, there are still vulnerabilities despite the introduction of new ITEF protocols designed to prevent large-scale attacks against public key infrastructures, which started to appear around 2011.

The use of SSL is in decline in favor of the newer and more secure TLS protocols. The security mechanisms presented within this research are outside of the TLS and SSL protocol definitions and therefore apply to both TLS and SSL. For clarity, this document uses TLS inclusively covering both TLS and SSL. This document explicitly calls out any differences in behavior between TLS and SSL.

1.1. The Historical Battleground

In parallel with the completion of the original CA/Browser forum standards, multiple security events unfolded which cast doubt on the sufficiency of the existing TLS security standards to protect TLS traffic.

1.1.1. Security Incident: Comodo

The first compromise of 2011 was to a Registration Authority (RA), the operator interface to a CA, affiliated with Comodo (Comodo, 2011). This attack resulted in nine rogue certificates. In this incident, detection and revocation of the rogue certificates occurred within hours (Stapleton, 2013). The root cause was the compromise of the affiliate's username and password, allowing full access to the CA for issuing rogue certificates. The investigation identified the invalid certificates in this attack.

1.1.2. Security Incident: Diginotar

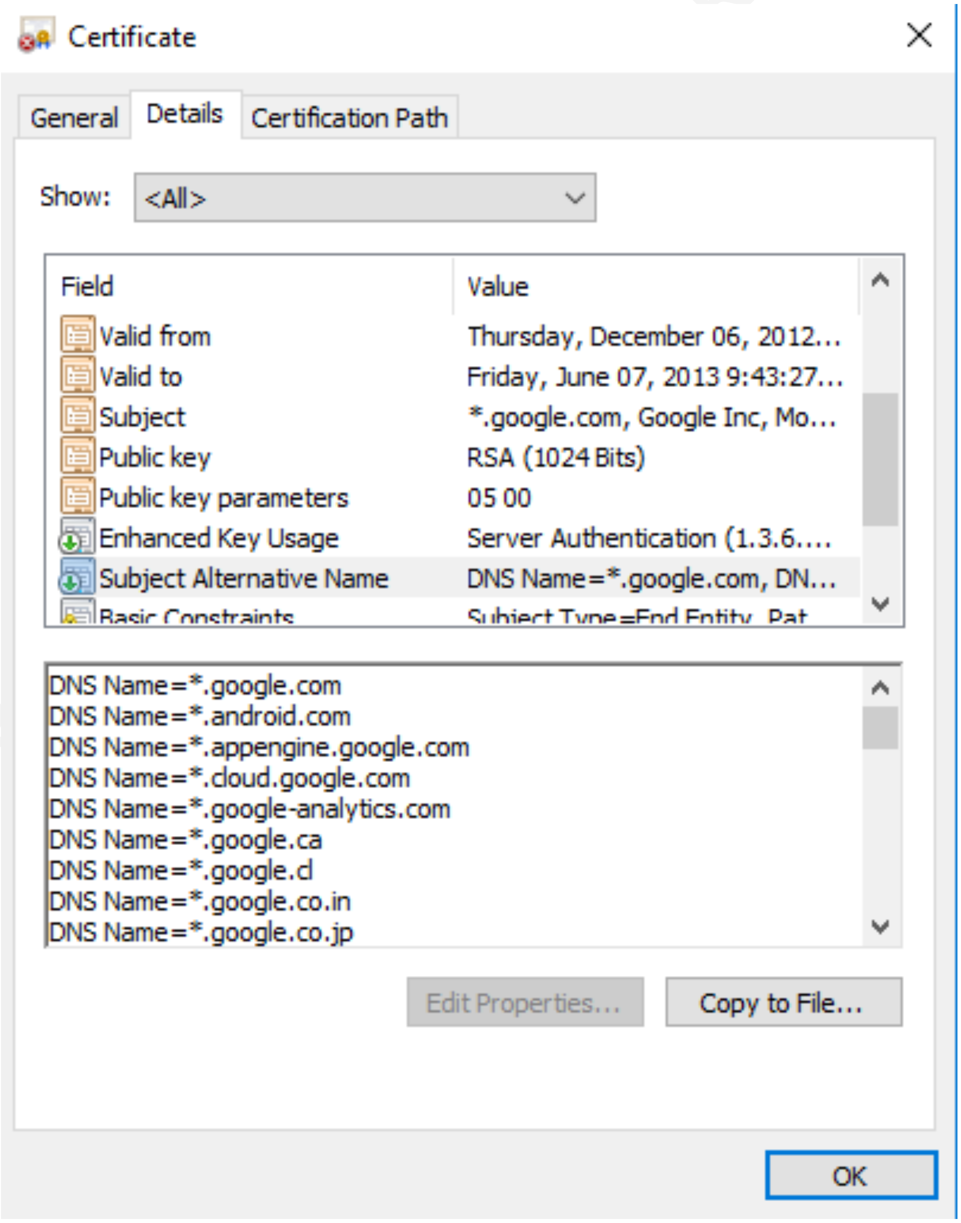
The second compromise in 2011 was a European-based CA service provider named Diginotar, which suffered a catastrophic security breach of multiple CAs resulting in the issuance of over 500 rogue TLS certificates (Prins, 2011). Certificates from this attack remained active for more than six weeks, and the attacker actively used them to compromise up to 300,000 Google accounts. The exploit implemented a large man-in-the-middle (MITM) attack using a rogue Google.com certificate targeting Iranian Internet users. The report describing the breach of the Certificate Authorities referred to as the Black Tulip Update, (Fox IT, 2012) can be downloaded from the Netherlands government archive. System security failures were the primary cause of this breach.

The reconnaissance period for this attack was completed by 17 June 2011 (Hein, 2013). The attack progressed towards the end goal over the following three weeks. Detection of the breach occurred on the 19th of July, and Diginotar revoked the identified rogue certificates, presuming containment by the end of July. There was a notification sent to Google on August 28th of a previously unknown rogue Google.com certificate as part of an ongoing attack against approximately 300,000 users mainly located in the Islamic Republic of Iran (Fox IT, 2012).

Evidence in the report shows that the attackers likely exploited a known vulnerability on an outdated software application and then used a username and password discovered on the external web server to attempt connections into a database server (Hein, 2013). This incident is an example of a malicious attack which persisted for an extended period. The investigation did not confirm the full scope of this attack.

1.1.3. Security Incident: Turk Trust

Again in 2011, the CA service provider Turk Trust issued two inappropriate CA certificates (Özarar, 2013). The recipient of the first certificate noticed the dangerous CA certificate extension and requested revocation by the CA. The Ankara Electric and Gas Operational Enterprise (EGO) received the second CA certificate and implemented it on their corporate webmail service. EGO exported the certificate and private key from the corporate webmail server in 2012 and imported them into their Checkpoint Firewall configured as a border TLS interception proxy. The EGO CA certificate remained in operation on the checkpoint firewall for four months. The firewall issued the rogue Google certificate shown in Figure 1 on June 12th, 2012. Turk Trust revoked the rogue Google certificate and the EGO CA certificate 20 days after the Checkpoint Firewall issued the rogue Google certificate.



Field	Value
Valid from	Thursday, December 06, 2012...
Valid to	Friday, June 07, 2013 9:43:27...
Subject	*.google.com, Google Inc, Mo...
Public key	RSA (1024 Bits)
Public key parameters	05 00
Enhanced Key Usage	Server Authentication (1.3.6....
Subject Alternative Name	DNS Name=*.google.com, DN...
Basic Constraints	Subject Type=End Entity, Pat

DNS Name=*.google.com
DNS Name=*.android.com
DNS Name=*.appengine.google.com
DNS Name=*.cloud.google.com
DNS Name=*.google-analytics.com
DNS Name=*.google.ca
DNS Name=*.google.cl
DNS Name=*.google.co.in
DNS Name=*.google.co.jp

Figure 1: Unauthorized Google.com certificate

This incident is an example of a non-malicious issuance of an inappropriate certificate. The firewall issued the certificate to a workstation internal to the EGO network as part of a data loss prevention or antivirus scanning of TLS secured data.

1.1.4. Security Incident: National Informatics Centre (NIC) of India

Hackers compromised CA servers run by the National Informatics Centre (NIC) of India in 2014 (Dong, Kane, & Camp, 2016). Rogue certificates issued for Google and Yahoo remained valid for nine days, at which point the India NIC revoked the intermediate CA (Langley, 2014). Similar to Diginotar, the root cause stated for this attack was the compromise of the CA issuance process resulting in four rogue certificates and, according to Google, the total number of invalid certificates was more than the reported quantity (Langley, 2014). Unlike Diginotar, the containment time for the incident was much shorter.

1.1.5. Security Incident: Symantec

In 2017, Google took the extraordinary step of black-listing Symantec CA Certificate Authorities due to a history of inappropriate certificate issuance. The root cause cited for revocation “that Symantec had entrusted several organizations with the ability to issue certificates without the appropriate or necessary oversight, and had been aware of security deficiencies at these organizations for some time“ (O'Brien, Sleevi, & Whalley, 2017). While not necessarily malicious, this incident identifies a significant vulnerability to TLS web services.

1.2. Potential for future incidents

The list of security incidents above supplies a broad range of incident types but is by no means complete. The compromise of a CA in the future is unpreventable. Even with the implementation of new TLS security mechanisms, there are still vulnerabilities. The development of new procedures will further reduce risk and, more importantly, quantify the residual risk for presentation to management for acceptance.

As a result of the significant attacks against CAs and the TLS protocols, web browser vendors supported the development of many new security protocols to correct the deficiencies. Proponents for each of these protocols are fighting for the browser vendors' support.

2. Research Method

The security incidents described in Section 1.0 cover a range of attacks and mistakes. The number of issued rogue certificates varies between two and over 500. The adversary intent ranges from accidental insider to malicious outsider. The exposure time ranged from hours to months. In two of the incidents, the investigators were unable to find all the rogue certificates issued. The analysis performed in this research will consider scalability, correctness, completeness, and robustness of the TLS mechanisms.

Scalability of TLS security mechanisms addresses attacks creating massive quantities of certificates and considers the ability of the protocol to protect more than just a small list of high-value domains, including Google and Facebook. Correctness looks at the functionality of the mechanism and addresses issues such as incorrect assumptions, self-denial of service, false positives, or false negatives. Completeness will look at whether or not the set of security mechanisms covers the full scope intended by the RFCs, while robustness assesses whether or not the system functions correctly with invalid inputs. A gap analysis of the lab system with improved TLS security mechanisms will highlight areas in need of procedural remediations or further security mechanisms. This analysis will concentrate on finding and quantifying the vulnerabilities which remain for both commercial and enterprise CAs with the security mechanisms in place.

Testing will include a lab setup of web servers (IIS and Apache) and browsers (Chrome, Firefox, Edge, and Internet Explorer) on both Windows and Linux hosts. The lab scope will include models of both public and internal Certificate Authorities. Testing of TLS security mechanisms will consist of full PKI integration for the IETF RFCs. Combining the complete set of TLS security mechanisms within the lab environment allows for a comprehensive approach to the gap analysis of the system.

Up-to-date browsers supporting DNSSEC, DANE TLSA, and CAA should find a rogue certificate from an unauthorized CA on the first connection to the TLS server.

Administrators should differentiate between rogue and valid certificates from an authorized CA based on a review of certificate transparency logs. Both public and internal CA implementations should supply equivalent results. Solutions must be scalable to millions of TLS sites in a fully automated manner. The research and experimentation will show that vendor support of TLS security mechanisms is still not complete and is not consistent among browser vendors. The well-implemented protocols require added procedures to be effective in mitigating risk. Addressing the identified gaps with processes or other mechanisms complements and completes the current best practices.

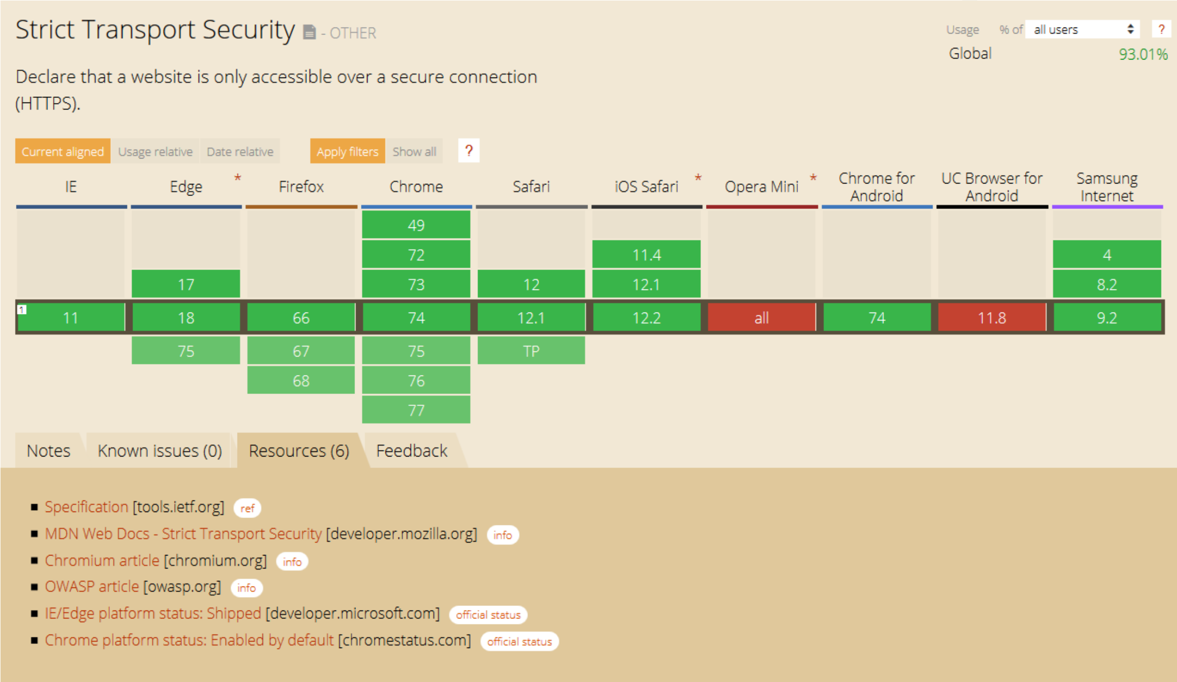
3. Findings and Discussion

In 2011, the CA /Browser forum adopted version 1.0 of its Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates. This document detailed the minimum process for confirming the identity of persons requesting certificates on behalf of an organization (CA/Browser Forum, 2011). Proof of identity was established based on the validation of Domain Name Registrar information for the subject TLS Certificate name or by “having the Applicant demonstrate practical control over the FQDN by making an agreed-upon change to information found on an online Web page identified by a uniform resource identifier containing the FQDN” (CA/Browser Forum, 2011). Later releases of the Baseline Requirement document have mandated the use of more TLS security mechanisms (CA/Browser Forum, 2014).

Each security mechanism supplies one or more security services. The following sections evaluate the effectiveness of these services and their vulnerabilities.

3.1. HTTP Strict Transport Security (HSTS)

As a first security mechanism defined in RFC6797, HSTS prevents external attackers from trying to downgrade web browser sessions from HTTPS to HTTP. Browser support for this protocol is already broadly proven, as shown in Figure 2. HSTS does not add more security services for TLS certificates themselves. Once a browser has received a valid HSTS record for a website, it will automatically redirect HTTP requests to HTTPS without requiring the server to respond with a redirect. The header includes an expiry time in a max-age value expressed in seconds. The header may also include a directive to include all subdomains. Browser vendors have also included an unofficial



Strict Transport Security - OTHER

Declare that a website is only accessible over a secure connection (HTTPS).

Usage % of all users ?
Global 93.01%

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			72		11.4				4
	17		73	12	12.1				8.2
11	18	66	74	12.1	12.2	all	74	11.8	9.2
	75	67	75	TP					
		68	76						
			77						

- Specification [tools.ietf.org] ref
- MDN Web Docs - Strict Transport Security [developer.mozilla.org] info
- Chromium article [chromium.org] info
- OWASP article [owasp.org] info
- IE/Edge platform status: Shipped [developer.microsoft.com] official status
- Chrome platform status: Enabled by default [chromestatus.com] official status

Figure 2: Browser support for HSTS *(Deveria, Can I use HSTS, 2019)*

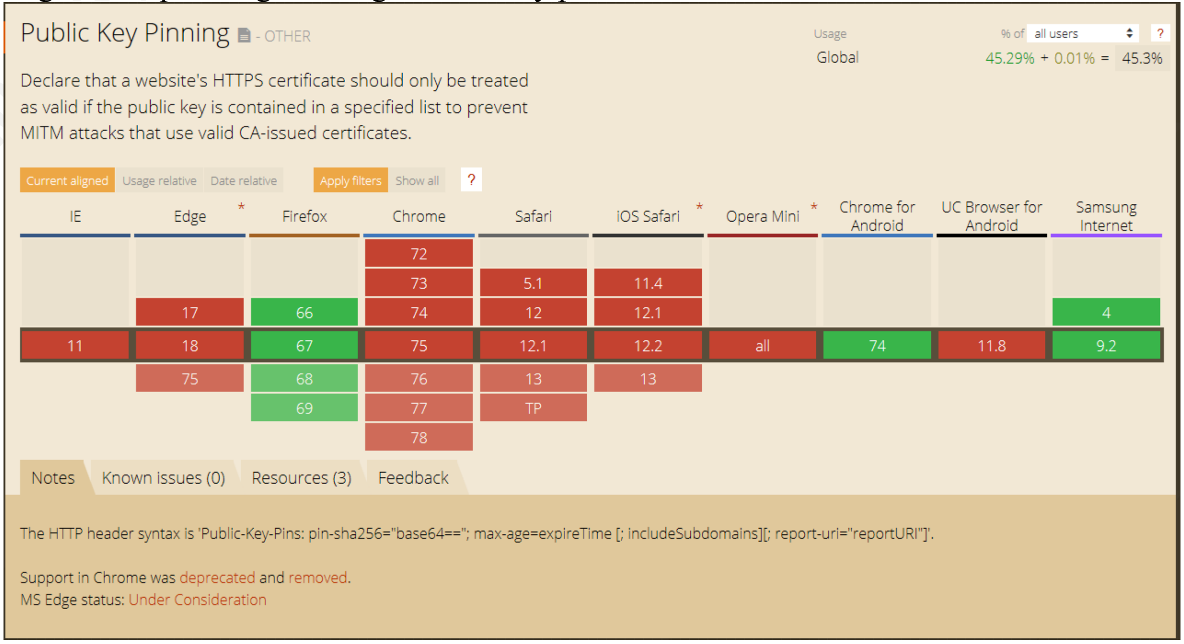
directive, used by web service providers to request preloading of the HSTS record by the browser vendors. HSTS aware web servers deliver an HSTS header in the HTTP response after TLS session negotiation. To specify mandatory TLS for the server and all subdomains for two years (63072000 seconds), the server would deliver "strict-transport-security: max-age=63072000; includeSubDomains" in the HTTP header.

Preloaded HSTS mitigates the risk of TLS stripping man-in-the-middle (MITM) attacks. While the MITM attack could remove the HSTS header information from the TLS server, it is unable to bypass the preloaded HSTS list in the browser. Once the browser has connected to the legitimate TLS service and has cached the HSTS header, it is immune to this MITM attack for the max-age of the cached information. The information in the HSTS header is stable for prolonged periods as it does not reference the keys used in TLS. The small size and long-life nature of HSTS headers make them robust and more scalable. Implementing HSTS in the test environment was straight forward and effective. Removal, by setting the maximum time to zero seconds, was also possible but required the TLS service to remain active until the earlier maximum timeframe had expired. If the web service vendor selected preloading by browser vendors (not tested), HSTS removal would be much more difficult.

3.2. HTTP Public Key Pinning (HPKP)

HPKP provides additional elements in an HTTP header which allows web service operators to bind a TLS public key to the site URL to reduce the incidence of MITM attacks caused by inappropriately-issued TLS certificates (Evans, Palmer, & Sleevi, 2015). The web service publishes the HPKP record during TLS negotiation.

HPKP, defined in RFC7469, did not supply a significant step forward in TLS security even though the large browser vendors, including Google, Apple, and Firefox adopted the protocol. Figure 3 details browser support for HPKP. Google contributed to the Internet Engineering Task Force (IETF)-proposed standard for HTTP Public Key Pinning (Evans, Palmer, & Sleevi, 2015), but are now in the process of removing pinning support (Palmer, 2019). Browser vendors have not confirmed the final date for removal. Accidental denial of service and the potential for malicious denial of service negatively offset the benefits of pinning. Migrating from an old key pair to a new key pair required significant planning. Pinning the new key pair into the HPKP header for an entire cache



Public Key Pinning - OTHER

Usage Global — % of all users: 45.29% + 0.01% = 45.3%

Declare that a website's HTTPS certificate should only be treated as valid if the public key is contained in a specified list to prevent MITM attacks that use valid CA-issued certificates.

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			72						
			73	5.1	11.4				
	17	66	74	12	12.1				4
11	18	67	75	12.1	12.2	all	74	11.8	9.2
	75	68	76	13	13				
		69	77	TP					
			78						

The HTTP header syntax is 'Public-Key-Pins: pin-sha256="base64=="; max-age=expireTime [; includeSubdomains][; report-uri="reportURI"]'.

Support in Chrome was deprecated and removed.
MS Edge status: Under Consideration

Figure 3: HPKP browser compatibility (Deveria, Can I use HPKP, 2019)

lifetime before the key rollover was critical. Caching errors causes a lockout of infrequent visitors from the website. Additionally, always having a backup key pair pin is crucial. Including pinning as a literature reference supplies completeness even though the browser manufactures dropped their commitments for HPKP.

Similar to HSTS, browser manufacturers include pre-installed HPKP lists in their products, but only for a smaller set of high-value domains. Users are not able to install pin lists manually into the browser. Pre-installed pin lists ensure that the pins name trustworthy keys, avoiding the scenario that the first connection happens to a compromised host or through a man-in-the-middle proxy. Pre-installed pin lists are not scalable to the millions of web service providers, as there are new certificates issued daily and a browser upgrade is the only mechanism to upgrade the pin list. Dynamic HPKP solves the scalability issue, but the browser assumes the first visit to the server (or the first visit after the cached information expires) is the legitimate connection without added verification by the browser. This assumption by the browser is known as a trust on first use. The assumed first security of a web site is a residual risk for HPKP correctness.

Beyond the limitations described above, a denial of service attack against HPKP is also a risk. If the web site operator does not properly manage TLS certificates within their dynamic HPKP header, the result is self-denial of service lasting for the duration of the HPKP max-age or until the restoration of the old public key to the server (Smashing Magazine, 2016).

Similarly, if an adversary adds a new HPKP header to a site or can alter keys and HKPK headers of an existing site, they may also cause a denial of service. If the maximum cache lifetime is high, this will result in a substantial denial of service to browsers visiting the site. Google currently caps the maximum cache time to 60 days, but this is still a long outage period. Figure 4 shows the calculation of the public key hash for www.giacenterprises.nl and the HPKP header with a 60-day (5184000 seconds) lifetime.

```
        openssl s_client -servername www.giacenterprises.nl -connect
www.giacenterprises.nl:443 | openssl x509 -pubkey -noout | openssl rsa -pubin -
outform der | openssl dgst -sha256 -binary | openssl enc -base64

        tMv3x8oVNiWt0EcbBqtVp7KSSw8ZAv9xDQ4xbNnK6vs=

Public-Key-Pins: pin-sha256=" tMv3x8oVNiWt0EcbBqtVp7KSSw8ZAv9xDQ4xbNnK6vs=";
max-age= 5184000
```

Figure 4: Extract hash of certificate public key and the HPKP header

Due to the denial of service issues described above, HPKP does not provide a robust solution to the TLS certificate vulnerabilities described in Section 2.0. Added to the scalability concerns and the assumed correctness on the first connection, HPKP did not meet its intended goal of TLS security.

3.3. Domain Name System Security Extensions (DNSSEC)

Several IETF RFCs including RFC4033, RFC4034, RFC4035, and RFC5011define Domain Name System Security Extensions (DNSSEC). These RFCs support several of the TLS security mechanisms described in this paper. Like public key infrastructures, DNSSEC implements a hierarchy of signatures from the top at the DNS root to the bottom at the end device record. Records within the DNSSEC database have resource record signatures (RRSIG). These signatures supply integrity to ensure that the records are not changed or spoofed during transmission. DNSSEC servers coexist with standard DNS servers, but DNSSEC does not work for a domain unless the parent of the domain is also DNSSEC-enabled. Implementing DNSSEC within an organization does not directly change the behavior of TLS implementations. Web browsers do not visualize DNSSEC information to the user without third-party browser extension software. These browser extensions are realistically only experimental, as browser version upgrades regularly affect their compatibility.

Decreasing the risk of DNS spoofing attacks is the primary goal of the Domain Name System Security Extensions. DNSSEC uses a hierarchical signing of records starting from the root, which signs all the well-known top-level domains. The top-level domains sign the subordinate entries within their authority, supplying a chain of trust back to the top. DNSSEC servers set up a hierarchical security association with each other. Web browsers do not provide rich native support for DNSSEC. The limited number of browser plugins supply varying levels of support to DNSSEC, and some plugin manufacturers have abandoned their projects. Figure 5 shows inaccurate information presented by the DNSSEC/TLSA Validator software from cz.nic labs. The errors are due to the DNS Root key rollover in the fall of 2018, which this product does not implement. Before 2018, this extension supported Internet Explorer, Chrome, and Firefox. Following changes to the Firefox API, cz.nic labs ceased development of the extension. The second plugin, HTTPS+ Checker for Firefox, shown in Figure 6, supplies correct DNSSEC information. Native end-to-end DNSSEC implementations, including DNSSEC aware web browsers, are not complete.

Regular debate occurs between security professionals about the viability and professed benefits of DNSSEC in comparison with other IETF RFCs such as HPKP, which does not rely on DNS or DNSSEC. Some perceive DNSSEC as resource-intensive, a privacy risk, and relying too much on the centralized control of the root keys. Privacy concerns have been addressed by enhancing the DNSSEC standard to support the prevention of zone walking (listing all the domain names in the zone). Centralized control concerns also exist for DNS without the DNSSEC extensions. DNSSEC performance impacts are a tradeoff against the added security preventing DNS poisoning.

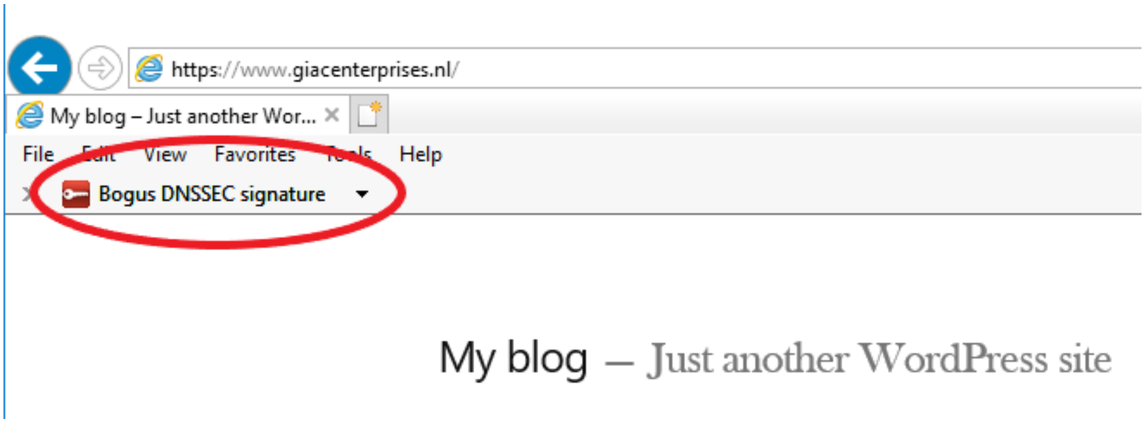

https://www.giacenterprises.nl/
My blog – Just another Wor...
File Edit View Favorites Tools Help
Bogus DNSSEC signature
My blog — Just another WordPress site

Figure 5: DNSSEC browser extension errors

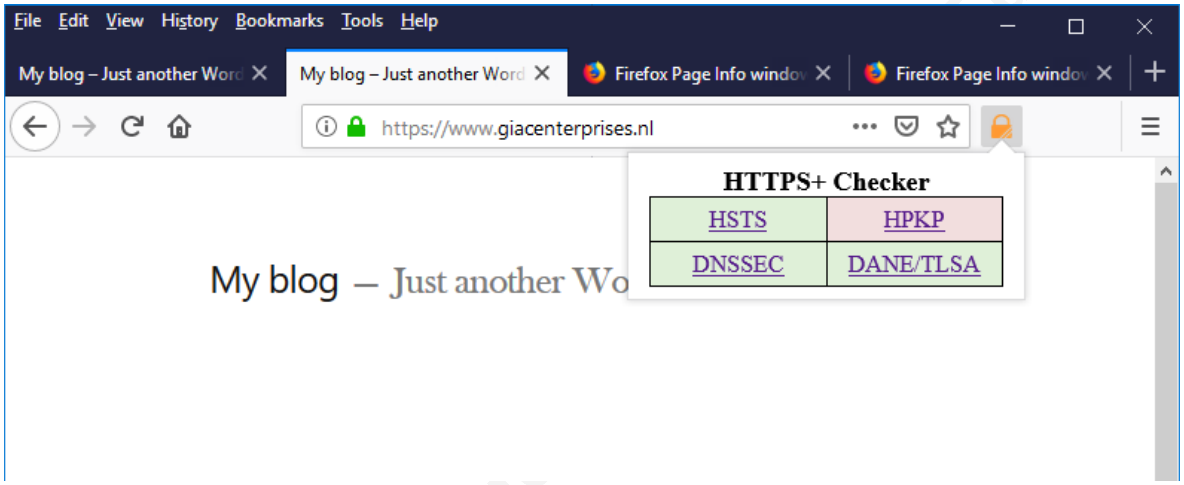

File Edit View History Bookmarks Tools Help
My blog – Just another Word
My blog – Just another Word
Firefox Page Info window
Firefox Page Info window
https://www.giacenterprises.nl
HTTPS+ Checker
HSTS
HPKP
DNSSEC
DANE/TLSA
My blog — Just another Wo

Figure 6: HTTPS+ Checker Plugin for Firefox

3.4. DNS-based Authentication of Named Entities (DANE)

DANE, specified in RFC6698, defines the TLSA resource record that binds a web service to either a key pair or an X.509-based certificate. Differing opinions also exist regarding the security and usability of these key pair types. Mandatory DNSSEC RRSIG signatures supply security for TLSA resource records, preventing spoofing during a DNS attack. DANE aware plugins perform DNSSEC lookups for the TLSA records and compare the certificate or public key information to the information presented by the web server during the TLS session negotiation. DANE TLSA records are similar to HPKP records in that they both bind the web service to the public keys presented by the web service. TLSA records differ from HPKP records in their delivery to the client and the security provided. TLSA records return through DNS queries. Updates to TLSA records will propagate from the DNS Server with a maximum latency of the DNS record life. Unlike HPKP, the DNS record life can be much smaller without affecting the security of the TLSA record.

DANE is another protocol which binds the public key or certificate to the web server. Instead of delivering this information to the browser within the HTTP header, DANE relies on DNSSEC to protect the information. Like HPKP, errors will result in service outages. However, DNS cache timeouts are much shorter than HPKP max-time values. Unlike HPKP, DANE is not vulnerable during the first connection to the web service. Secure installation of the root DNS keys occurs with the DNSSEC aware resolvers. Like DNSSEC, native end-to-end DANE implementations, including browsers, are not complete.

3.5. DNS CA Authorization (CAA) Resource Records

Unlike TLSA records, which supply information to the browser regarding valid certificates, CAA records identify the subject domain and the authoritative CA to issue certificates (Hallam-Baker & Stradling, 2013). Sample CAA records in Figure 7 explain the record type and function. CAA also includes an "Issuer critical" flag to indicate that the CA must understand the data to complete the transaction. Setting this flag to 0 allows the CA to ignore the record if it does not understand the semantics. Records include "issue" and "issuewild" tags to show whether certificates permit a wildcard subject, or just for the defined host. The optional "iodef" tag supplies a URL for reporting inconsistencies or policy violations.

<table>
<tr><th>Subject Domain</th><th>Type</th><th>Issuer Critical</th><th>Tag</th><th>Value</th></tr>
<tr><td>giacenterprises.nl.</td><td>CAA</td><td>0</td><td>issuewild</td><td>"digicert.com"</td></tr>
<tr><td colspan="5">Wild card certificate allowed for *.giacenterprises.nl from the Digicert CA.</td></tr>
<tr><td>giacenterprises.nl.</td><td>CAA</td><td>0</td><td>issue</td><td>"digicert.com"</td></tr>
<tr><td colspan="5">Certificate allowed for giacenterprises.nl from the Digicert CA.</td></tr>
<tr><td>giacenterprises.nl.</td><td>CAA</td><td>0</td><td>iodef</td><td>mail@giacenterprises.nl</td></tr>
<tr><td colspan="5">CA may report invalid certificate requests to mail@giacenterprises.nl</td></tr>
<tr><td>giacenterprises.nl.</td><td>CAA</td><td>128</td><td>issue</td><td>"digicert.com; policy=ev"</td></tr>
<tr><td colspan="5">The DigiCert CA must understand the issuance policy for Extended Validation certificates before issuing certificates for giacenterprises.nl</td></tr>
</table>

Figure 7: Example CAA resource records

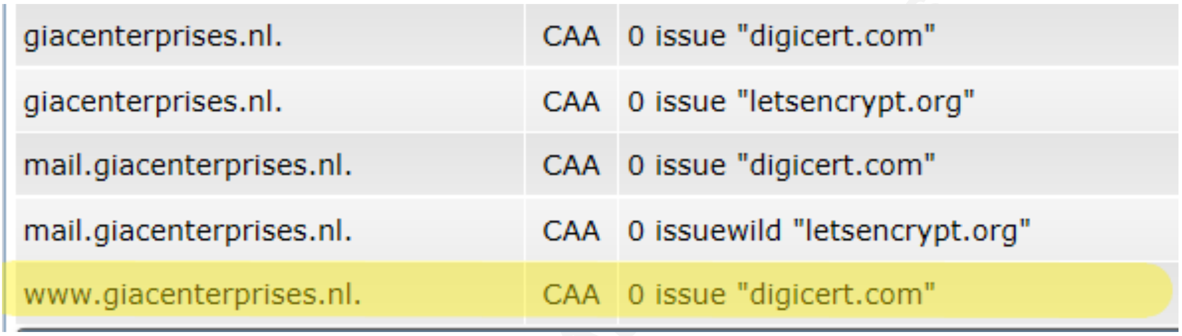
Public CA that are compliant with the CA/Browser Forum Certificate policy must comply with CAA resource records (CA/Browser Forum, 2014). The standard does not specify validation methods or protocols for the CAA records. CA service providers enforce the certificate issuance policy either through procedural processes or via CAA-aware protocol implementations such as the Automated Certificate Management Environment (ACME) protocol defined in RFC8555 (Barnes, Hoffman-Andrews, McCarney, & Kasten, 2019). While not mandatory in the RFC, DNSSEC resource record signatures supply protection for DNS CAA resource records.

Each certificate supports multiple subject domains. The subject domain may be present as the common name for the certificate, or in the Subject Alternative Name extension within the certificate. The rules for CAA record verification apply to both common name and certificate extensions.

The fundamental vulnerability not addressed by CAA records is the rogue or hacked Certificate Authority which will issue certificates despite the contents of the CAA records. There is currently no support within browsers for validating CAA records against already-issued web server certificates. Positive and negative testing for CAA record validation supplied the expected results. The first stage of testing relied on web-based tools to evaluate CAA records to determine their validity for certificate issuance (Internet Security Research Group, 2019) from the Let's Encrypt CA.

The first baseline test included correct CAA records for the Let's Encrypt CA. The hosting service provider implements the ACME protocol for certificate issuance. The certificate request sent via the web hosting provider for the giacenterprises.nl site returned successfully. The second test required a CAA record with only an alternate CA included for one of the subject domains. The certificate request included three domain names, including giacenterprises.nl, www.giacenterprisess.nl, and mail.giacenterprises.nl. In Figure 8, www.giacenterprises.nl does not have a CAA record for the Let's Encrypt CA. The web-based CAA validation tool supplied the error message shown in Figure 9.

The next test published CAA records prohibiting the Let's Encrypt CA from issuing certificates for mail.giacenterprises.nl. The web-based CAA validation tool for this scenario showed no errors, despite the inclusion of the prohibited certificate extension mail.giacenterprises.nl entry in the request. The resulting certificate, correctly issued without the disallowed DNS name, appears highlighted in Figure 10.

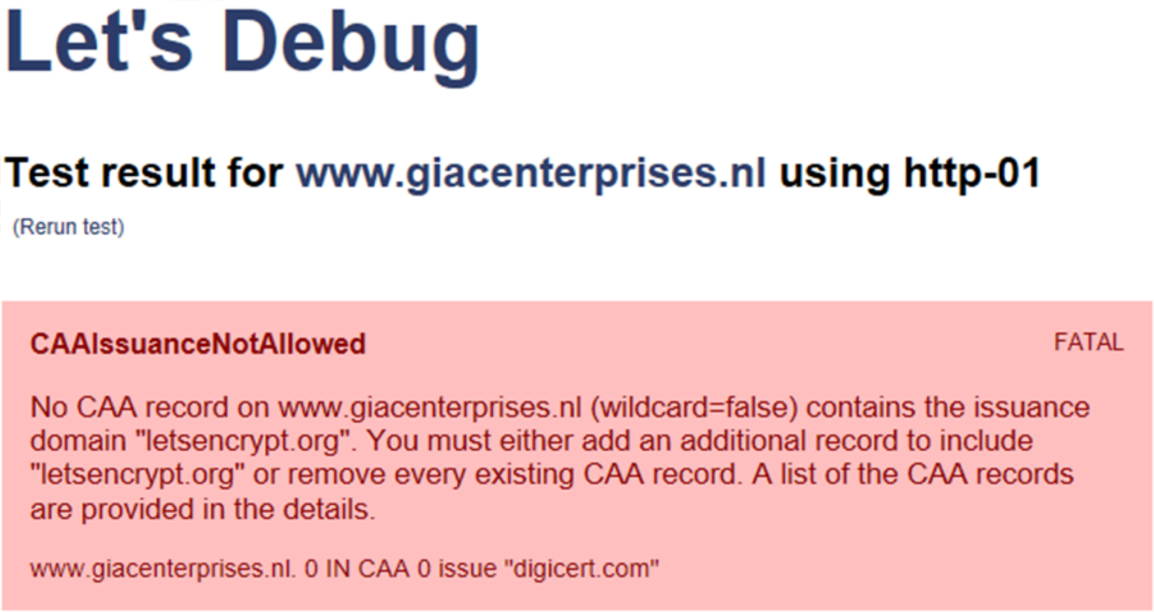


giacenterprises.nl.	CAA	0 issue "digicert.com"
giacenterprises.nl.	CAA	0 issue "letsencrypt.org"
mail.giacenterprises.nl.	CAA	0 issue "digicert.com"
mail.giacenterprises.nl.	CAA	0 issuewild "letsencrypt.org"
www.giacenterprises.nl.	CAA	0 issue "digicert.com"

Figure 8: DNS CAA record for inappropriate SAN test

3.6. Certificate Transparency (CT)

The concept of certificate transparency includes Signed Certificate Timestamp (SCT) records published into a set of log files. Subsequently, these records cannot be removed or altered. Auditing these log files confirms that only valid certificates have been issued for a domain to verify that SCT records presented by a TLS service match with a set of CT log *servers.*



Let's Debug

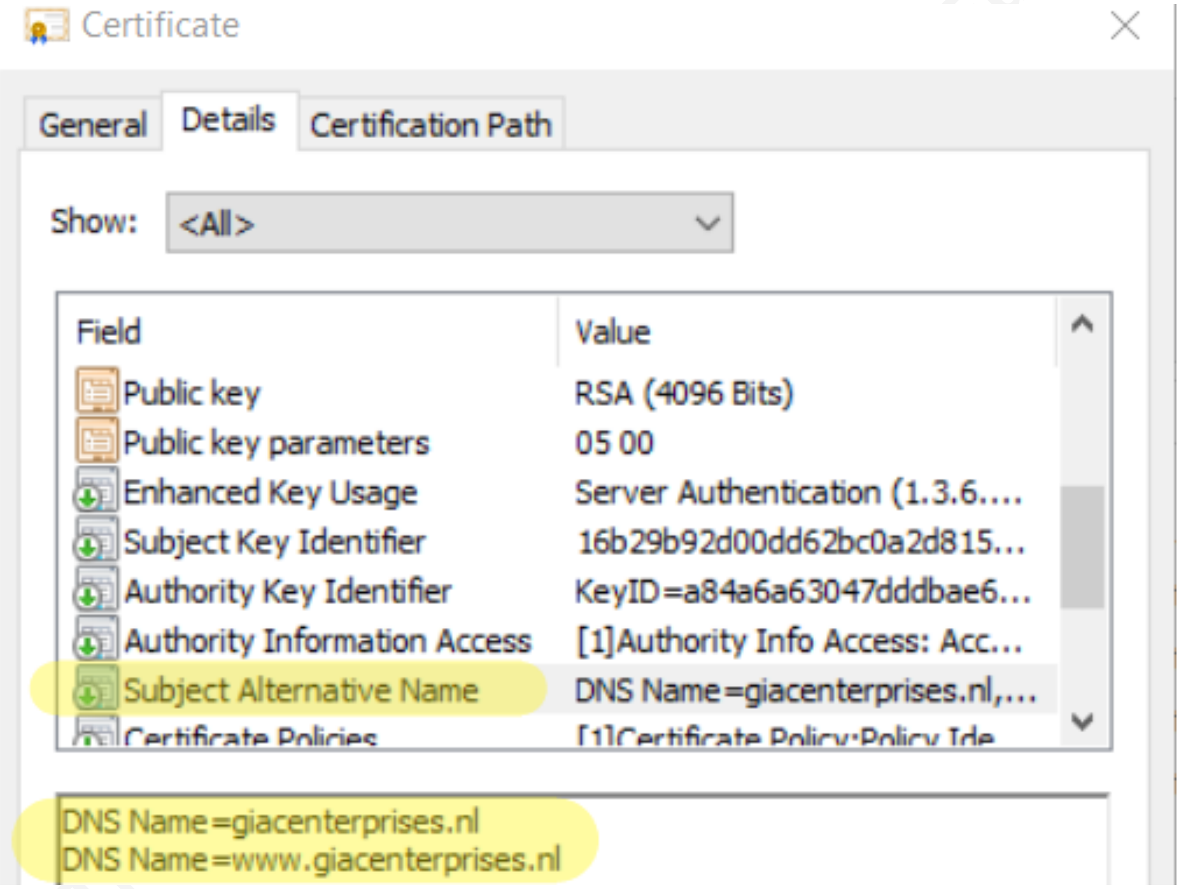
Test result for www.giacenterprises.nl using http-01

CAAIssuanceNotAllowed FATAL

No CAA record on www.giacenterprises.nl (wildcard=false) contains the issuance domain "letsencrypt.org". You must either add an additional record to include "letsencrypt.org" or remove every existing CAA record. A list of the CAA records are provided in the details.

```
www.giacenterprises.nl. 0 IN CAA 0 issue "digicert.com"
```

Figure 9: Let’s Encrypt CAA record not found for www.giacenterprises.nl



Field	Value
Public key	RSA (4096 Bits)
Public key parameters	05 00
Enhanced Key Usage	Server Authentication (1.3.6....
Subject Key Identifier	16b29b92d00dd62bc0a2d815...
Authority Key Identifier	KeyID=a84a6a63047dddbae6...
Authority Information Access	[1]Authority Info Access: Acc...
Subject Alternative Name	DNS Name=giacenterprises.nl,...
Certificate Policies	[1]Certificate Policy:Policy Ide

DNS Name=giacenterprises.nl
DNS Name=www.giacenterprises.nl

Figure 10: Certificate issued without unauthorized DNS Name

Certificate transparency is an experimental RFC. Chrome, Apple, and Firefox are committed to implementing CT. Chrome enforces certificate transparency for all certificates issued after April 2018 and Safari enforcement started for certificates issued after October 15, 2018. While a plugin is available for Firefox, it only validates SCT records embedded within the web server certificate. The Developer tools menu in the Chrome browser allows for the verification of Signed Certificate Timestamp (SCT) data against the log server information found within the certificate or the TLS handshake.

Figure 11 shows the high-level protocol exchange for including an SCT record into a certificate. Instead of supplying a signed certificate to the web server at once, the CA first provides a precertificate which the client or CA forwards to the CT log server.

The CA signs the final certificate, including the SCT record returned from the CT log servers. Alternatively, the web server may send SCT records to the client during the TLS handshake.

Certificate authorities do not register SCT information in all CT log servers. The residual risks here are broad since CT enforcement currently occurs on a portion of TLS certificates. Certificates issued before April 2018 are not required to implement CT, and browser support for CT is still far from complete. CT validation in the browser requires specific user activities and understanding of the protocol. For the web service owner, the absence of malicious CT log entries does not guarantee TLS certificate security.

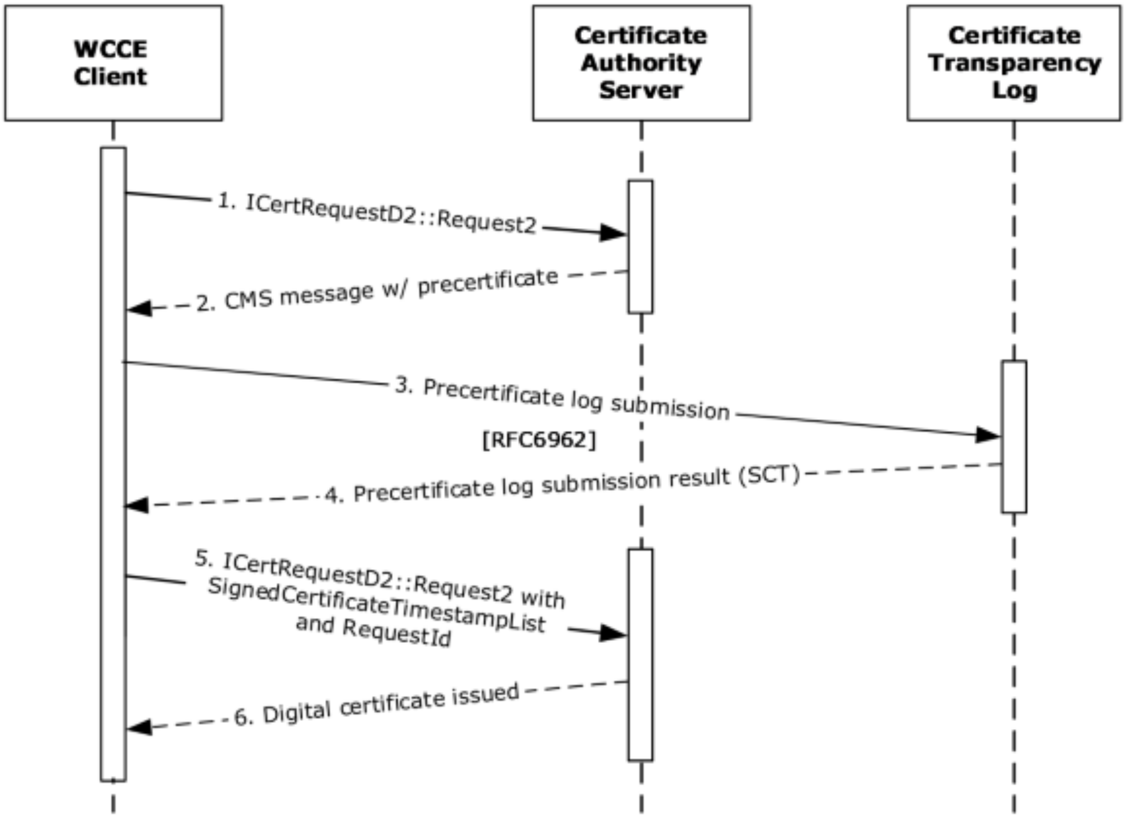
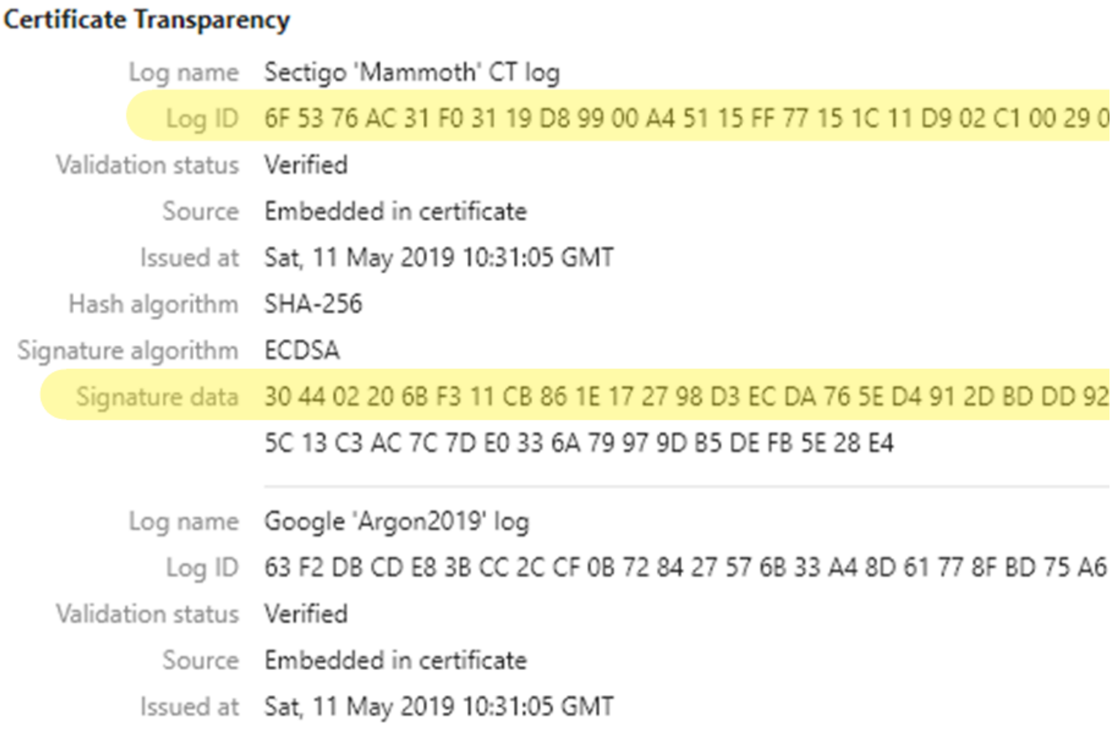

WCCE Client
Certificate Authority Server
Certificate Transparency Log
1. ICertRequestD2::Request2
2. CMS message w/ precertificate
3. Precertificate log submission
[RFC6962]
4. Precertificate log submission result (SCT)
5. ICertRequestD2::Request2 with SignedCertificateTimestampList and RequestId
6. Digital certificate issued

Figure 11: Certificate issued with certificate transparency (Microsoft, 2019)

CT log service providers require more maturity, and browser vendors must add full support for CT. Searching the Comodo CT log for “google.com” resulted in more than 1300 responses. Searching for “google” resulted in a crash of the user interface.

The Chrome browser SCT information for the www.giacenterprises.nl domain appears in Figure 12. This figure shows the CT log server for each of the two SCT fields. Each SCT signature confirms the correctness of the entries and their timestamps. No certificate chain information exists within these entries. Validation by the browser requires a lookup of the web server identity within the CT log to find the corresponding certificate and SCT entry. Comparison with the SCT information in the certificate shows the same Log ID and Signature data in Figure 13 but without the header information.

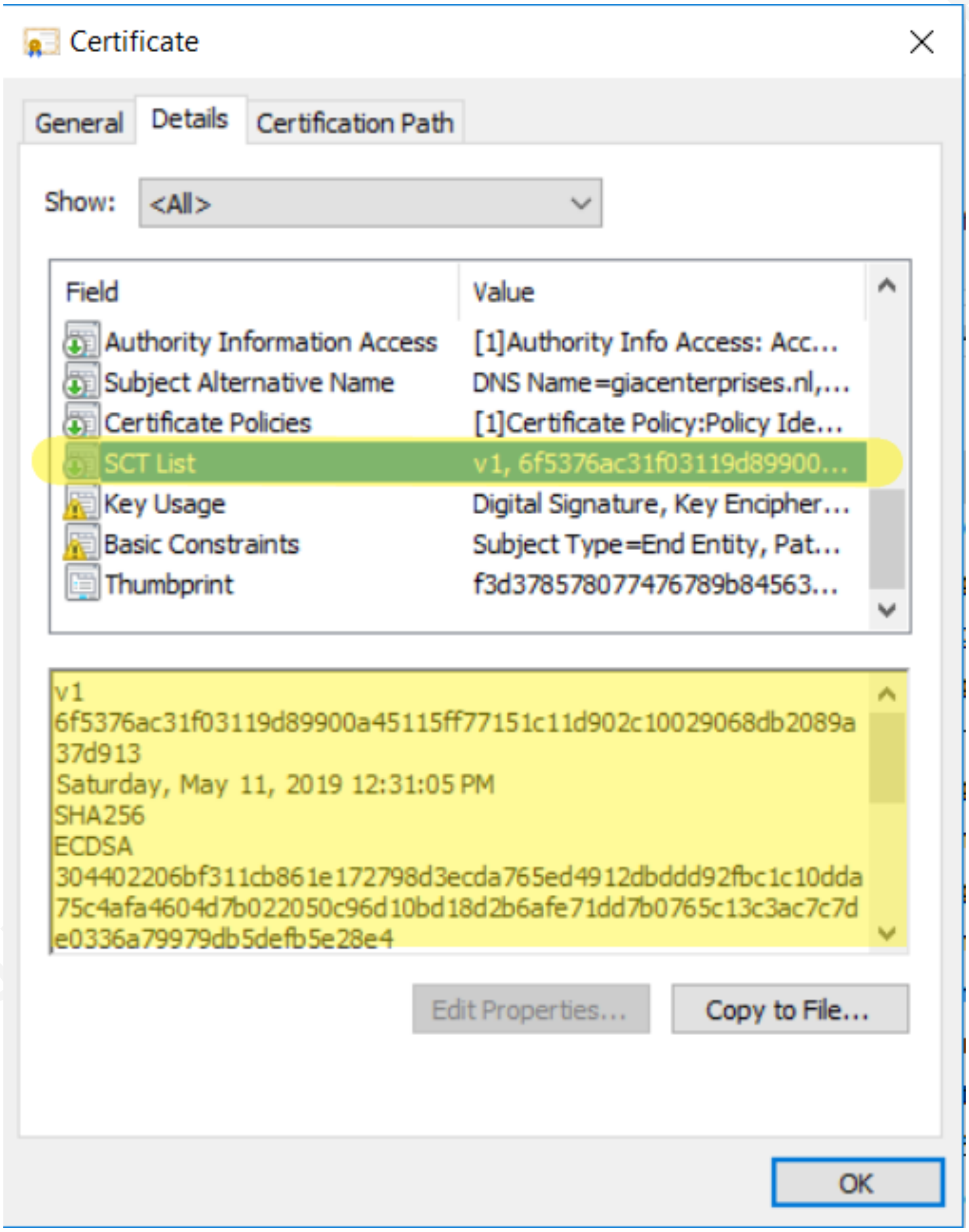


Certificate Transparency

Log name	Sectigo 'Mammoth' CT log
Log ID	6F 53 76 AC 31 F0 31 19 D8 99 00 A4 51 15 FF 77 15 1C 11 D9 02 C1 00 29 0
Validation status	Verified
Source	Embedded in certificate
Issued at	Sat, 11 May 2019 10:31:05 GMT
Hash algorithm	SHA-256
Signature algorithm	ECDSA
Signature data	30 44 02 20 6B F3 11 CB 86 1E 17 27 98 D3 EC DA 76 5E D4 91 2D BD DD 92 5C 13 C3 AC 7C 7D E0 33 6A 79 97 9D B5 DE FB 5E 28 E4
Log name	Google 'Argon2019' log
Log ID	63 F2 DB CD E8 3B CC 2C CF 0B 72 84 27 57 6B 33 A4 8D 61 77 8F BD 75 A6
Validation status	Verified
Source	Embedded in certificate
Issued at	Sat, 11 May 2019 10:31:05 GMT

Figure 12: Chrome browser listing of SCT entries for giacenterprises.nl

Search results vary between logs, and even between web-based log search tools. The Digicert CT monitor (https://ssltools.digicert.com/checker/views/ctsearch.jsp) does not locate the certificate for www.giacenterprises.nl, encoded as a Subject Alternative Name (SAN) DNS entry within the certificate. A later search for giacenterprises.nl provided no results other than a "server not available" error message.



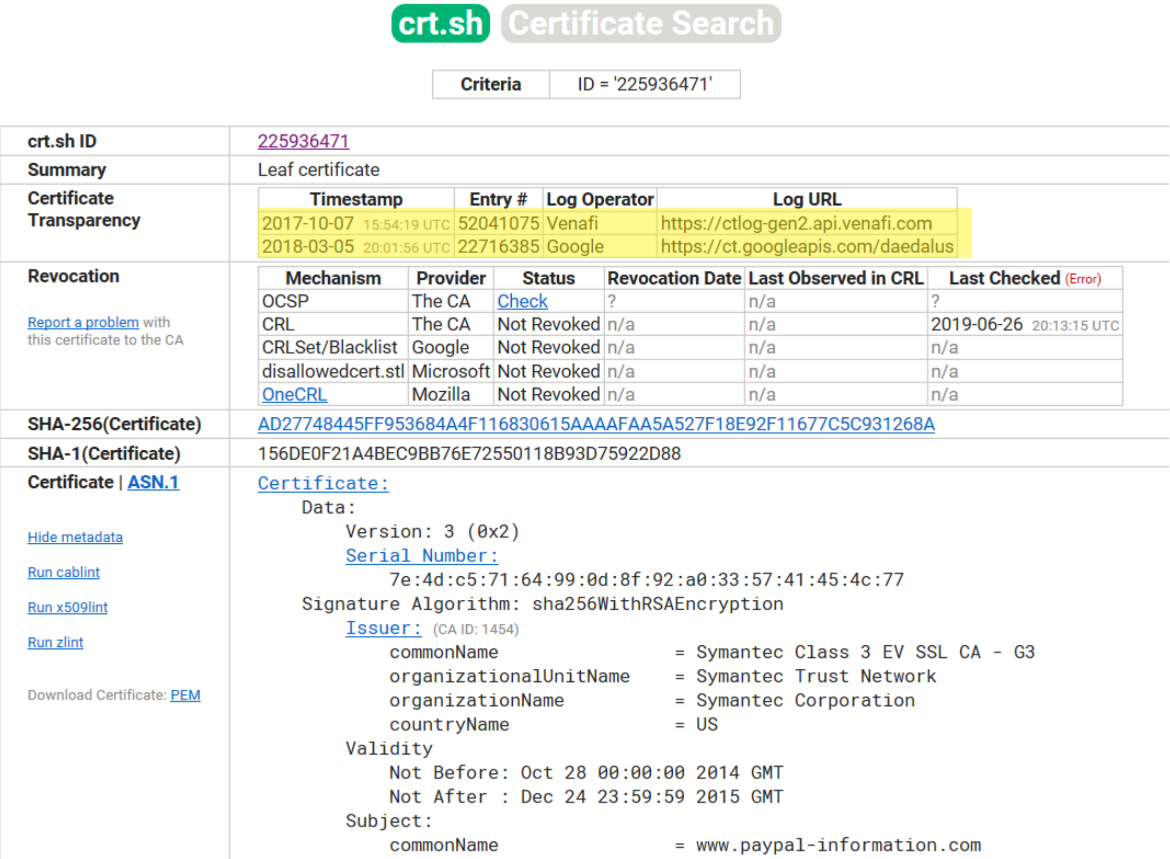
Show: <All>

Field	Value
Authority Information Access	[1]Authority Info Access: Acc...
Subject Alternative Name	DNS Name=giacenterprises.nl,...
Certificate Policies	[1]Certificate Policy:Policy Ide...
SCT List	v1, 6f5376ac31f03119d89900...
Key Usage	Digital Signature, Key Encipher...
Basic Constraints	Subject Type=End Entity, Pat...
Thumbprint	f3d378578077476789b84563...

```
v1
6f5376ac31f03119d89900a45115ff77151c11d902c10029068db2089a
37d913
Saturday, May 11, 2019 12:31:05 PM
SHA256
ECDSA
304402206bf311cb861e172798d3ecda765ed4912dbddd92fbc1c10dda
75c4afa4604d7b022050c96d10bd18d2b6afe71dd7b0765c13c3ac7c7d
e0336a79979db5defb5e28e4
```

Figure 13: SCT entries in certificate for giacenterprises.nl

The CT monitor operated by Entrust Datacard (https://www.entrust.com/ct-search/) shows 38 valid certificate entries in response to a search for PAYPAL.JP with subdomains included. The CT monitor hosted by Sectigo Limited (https://crt.sh) shows 52 certificate entries in response to a search for %PAYPAL.JP. The % character is a wildcard indicator to include subdomains. Both CT monitors supply search results based on the common name and SAN entries. The Entrust search engine does not find certificates which have the Subject Alternative Names capitalized regardless of the case used in the search query. Entrust also did not find one certificate (Figure 14) logged in the Venafi and Google Daedalus CT logs.



crt.sh Certificate Search

Criteria	ID = '225936471'

crt.sh ID	225936471
Summary	Leaf certificate

Certificate Transparency

Timestamp	Entry #	Log Operator	Log URL
2017-10-07 15:54:19 UTC	52041075	Venafi	https://ctlog-gen2.api.venafi.com
2018-03-05 20:01:56 UTC	22716385	Google	https://ct.googleapis.com/daedalus

Revocation

Report a problem with this certificate to the CA

Mechanism	Provider	Status	Revocation Date	Last Observed in CRL	Last Checked (Error)
OCSP	The CA	Check	?	n/a	?
CRL	The CA	Not Revoked	n/a	n/a	2019-06-26 20:13:15 UTC
CRLSet/Blacklist	Google	Not Revoked	n/a	n/a	n/a
disallowedcert.stl	Microsoft	Not Revoked	n/a	n/a	n/a
OneCRL	Mozilla	Not Revoked	n/a	n/a	n/a

SHA-256(Certificate)	AD27748445FF953684A4F116830615AAAAFAA5A527F18E92F11677C5C931268A
SHA-1(Certificate)	156DE0F21A4BEC9BB76E72550118B93D75922D88

Certificate | ASN.1

Hide metadata

Run cablint

Run x509lint

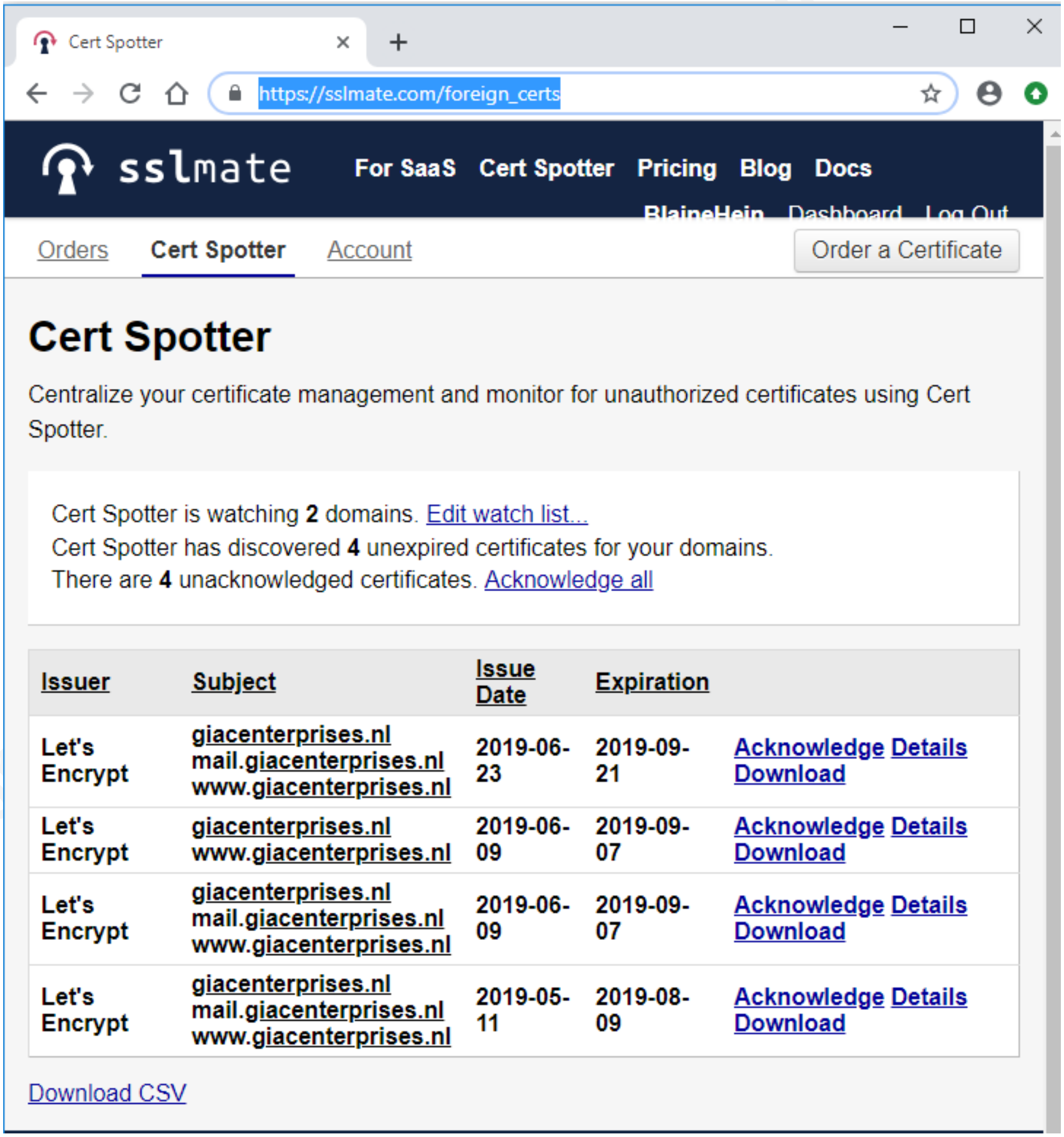
Run zlint

Download Certificate: PEM

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            7e:4d:c5:71:64:99:0d:8f:92:a0:33:57:41:45:4c:77
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: (CA ID: 1454)
            commonName                = Symantec Class 3 EV SSL CA - G3
            organizationalUnitName    = Symantec Trust Network
            organizationName          = Symantec Corporation
            countryName               = US
        Validity
            Not Before: Oct 28 00:00:00 2014 GMT
            Not After : Dec 24 23:59:59 2015 GMT
        Subject:
            commonName                = www.paypal-information.com
```

Figure 14: Certificate only found in the Sectigo CT Monitor

Subscription to a log monitoring service is another possibility for CT monitoring. There are several choices for monitoring services. As an example, the sslmate Cert Spotter service (https://sslmate.com/certspotter/) results appear in Figure 15.



Cert Spotter

Centralize your certificate management and monitor for unauthorized certificates using Cert Spotter.

Cert Spotter is watching **2** domains. Edit watch list...
Cert Spotter has discovered **4** unexpired certificates for your domains.
There are **4** unacknowledged certificates. Acknowledge all

Issuer	Subject	Issue Date	Expiration	
Let's Encrypt	giacenterprises.nl mail.giacenterprises.nl www.giacenterprises.nl	2019-06-23	2019-09-21	Acknowledge Details Download
Let's Encrypt	giacenterprises.nl www.giacenterprises.nl	2019-06-09	2019-09-07	Acknowledge Details Download
Let's Encrypt	giacenterprises.nl mail.giacenterprises.nl www.giacenterprises.nl	2019-06-09	2019-09-07	Acknowledge Details Download
Let's Encrypt	giacenterprises.nl mail.giacenterprises.nl www.giacenterprises.nl	2019-05-11	2019-08-09	Acknowledge Details Download

Download CSV

Figure 15: Certificate Monitoring service from sslmate

3.7. Internal TLS interception vulnerabilities

Within an enterprise, TLS interception proxies supply a part of a data loss prevention program. The use of TLS interception proxies adds residual risk to the organization. The only certificates presented to the browsers are the certificates from servers inside the organization, and the interception certificates issued by the proxy server. Since proxy certificates do not match a publicly-trusted CA, HPKP, certificate transparency, CAA records, and DNSSEC are either disabled by the browser or supply no added security to the browser. The responsibility for validating external web servers remains with the TLS interception proxy.

4. Recommendations and Implications

The recommendations below result from the current immaturity of TLS security standards and the incomplete implementations from browser vendors. Effective security implementations require native support within the browser instead of third party addons. TLS security protocols lacking robustness or scalability are not recommended.

4.1. Recommendations for Practice

4.1.1. HTTP Public Key Pinning (HPKP)

Organizations should not implement HPKP primarily due to the potential for self-denial of service, and its impending removal from browsers.   In line with the CA/Browser Forum guidelines, they should implement certificate transparency and CAA records instead.

4.1.2. HTTP Strict Transport Security (HSTS)

Any organization with an internal PKI should plan HSTS carefully to ensure that HTTP CRL Distribution Points and OCSP responders are still accessible. These services must use HTTP. Validating the security of a certificate through a TLS channel is not possible without requiring a TLS connection, which in turn needs the validation of a certificate.

4.1.3. Domain Name System Security Extensions (DNSSEC)

DNSSEC should be deployed to protect against DNS attacks. As the DNS keys are hierarchical, there is no need to set long TTL values to avoid trust on first use vulnerabilities. Web service providers should confirm the completeness of their DNSSEC configuration to ensure that the DNS service supports RRSIG signatures for all required resource records.

4.1.4. DNS-based Authentication of Named Entities (DANE)

Despite the current lack of broad support, DANE TLSA resource records could supply a potential replacement for HPKP in the future if browser manufacturers add support. However, organizations should use certificates from a publicly-trusted CA server that is part of the CA/Browser Forum for TLSA records. The CT Log servers do not support certificates or key pairs issued by private Certificate Authorities to avoid spam.

4.1.5. DNS CA Authorization (CAA) Resource Records

Web service providers should use CAA resource records. When using certificate transparency, the CAA record will need to specify a CA/Browser forum public CA server. While CAA records will not protect against a rogue CA, they will decrease the exposure to accidental certificate issuance. Any CA completing its due diligence before certificate issuance will have access to the CAA record, thereby preventing certificate issuance by a random CA. Vigilance in monitoring certificate transparency is still required as CAA record validation may be procedural instead of being enforced by the CA system software.

4.1.6. Certificate Transparency

Certificates for public-facing TLS servers should include certificate transparency. Web Service owners should consult multiple CT Monitors to decrease the exposure to unauthorized certificates. Web service providers must watch all subdomains, regardless of their physical existence. Large multinational organizations should watch across all top-level DNS domains to prevent domain squatting attacks. Certificate transparency requires web service owner vigilance, either through scanning all CT log servers or through a subscription to a scanning service.

4.1.7. Internal TLS interception

TLS interception proxies should never use certificates from a public PKI, or an organization's internal CA. Issue TLS interception proxy certificates from an offline CA dedicated to the TLS interception proxy.

Organizations must test their TLS interception proxies to ensure that they enforce proper TLS security mechanisms. As a minimum, the proxy must perform the validation ordinarily completed by the Browser when accessing web services secured with certificates from a public CA. The proxy must validate SCT records for external web sites visited by the browser. The proxy must support CRLs and OCSP for validating the web server certificate chains. The trusted root CA store of the proxy must match the CA/Browser forum root list. Proxies must send HSTS headers to the web browsers.

4.2. Residual risk and other security recommendations

During testing, two areas not addressed by the TLS security mechanisms in this paper stood out regarding CA business processes.

4.2.1. Short life certificates and reuse of private keys

Certificate Authorities such as Let's Encrypt adopted a new trend, issuing short life credentials claiming reduced importance for certificate revocation checking at the same time as decreasing the duration of a successful exploit on TLS Web services. Unfortunately, there is also a tendency to perform HPKP or DANE TLSA records based on the Subject Public Key Information (SPKI). Later certificate requests are then based on the same key pair to reduce workload and potential impact of changing the relevant records. The risk here is that renewing a certificate based on the same key pair does not break any system compromise, as the attacker needs only to load the new certificate to continue their exploit. Updated certificates should always use a new key pair regardless of certificate lifetime.

4.2.2. Wild card certificates

The excessive use of wildcard certificates exposes a much larger community to the risk of a certificate compromise. Compromise of a certificate for *.giacenterprises.nl puts all hosts in the organization at risk. Even when subdomain web servers do not use wildcard certificates, they are still vulnerable to a MITM attack using the wildcard certificate.

4.3. Implications for Future Research

There are three recommended areas for future research. First, an investigation into the effectiveness of alternatives to TLS interception proxies to further research published in The Sorry State of TLS Security in Enterprise Interception Appliances (Wakek, Mannan, & Youssef, 2018). The second area for future research is the correctness of the ACME protocol and its implementations for CA/Browser Forum compliance. The final area of research recommended is the validation of CAA records by browsers to increase the detection of rogue certificates before use.

The weakest area of test results found within this report was the monitoring of certificate transparency logs. Future testing of correctness and completeness of the various CT monitoring web services and CT monitoring service providers would improve their robustness.

5. Conclusion

The implementation of up-to-date TLS security mechanisms allows for rapid detection of rogue certificates and a decreased likelihood of accidental issuance of rogue certificates. CAA resource records control certificate issuance, reducing exposure to compromised CAs. If fully implemented, DANE provides an added layer of security binding the TLS public key to the website, reducing exposure to unauthorized certificates from any CA. DNSSEC decreases DNS poisoning attacks against web services and browsers that are fully DNSSEC aware.

While the certificate transparency results presented show promise, they still rely on the discovery of rogue certificates after compromise. Proactively detecting rogue certificates before their use in an attack requires full support by the browser vendors for DANE and DNSSEC. In addition to CAA record adoption by Certificate Authorities, browser vendors should consider validating TLS Server certificates against CAA records during TLS negotiation.

References

Barnes, R., Hoffman-Andrews, J., McCarney, D., & Kasten, J. (2019, March). *RFC8555 Automatic Certificate Management Environment (ACME)*. Retrieved from Internet Engineering Task Force.

CA/Browser Forum. (2011, November 22). *Baseline Requirements Documents.* Retrieved from CA/Browser Forum: https://cabforum.org/wp-content/uploads/Baseline_Requirements_V1.pdf

CA/Browser Forum. (2014, April 14). *Baseline Requirements Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates.* Retrieved from CA/Browser Forum: https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.4.5.pdf

Comodo. (2011, March 23). *Comodo SSL Affiliate The Recent RA Compromise*. Retrieved from COMODO CYBERSECURITY: https://blog.comodo.com/other/the-recent-ra-compromise/

Deveria, A. (2019, 05 28). *Can I use HPKP*. Retrieved from Can I use: https://caniuse.com/#feat=publickeypinning&feature_sort=score

Deveria, A. (2019, May 28). *Can I use HSTS*. Retrieved from Can I use: https://caniuse.com/#feat=stricttransportsecurity&feature_sort=score

Dong, Z., Kane, K., & Camp, L. J. (2016, September). Detection of Rogue Certificates from Trusted Certificate Authorities Using Deep Neural Networks. *ACM Transactions on Privacy and Security, Vol. 19, No. 2, Article 5*, 31. Retrieved from https://www.researchgate.net/publication/316940647_Detection_of_Rogue_Certificates_from_Trusted_Certificate_Authorities_Using_Deep_Neural_Networks

Evans, C., Palmer, C., & Sleevi, R. (2015, April). *Public Key Pinning Extension for HTTP.* Retrieved from Internet Engineering Task Force (IETF): https://tools.ietf.org/html/rfc7469

Fox IT. (2012, August 13). *Black Tulip Update.* Retrieved from Archiefweb.eu: https://archief06.archiefweb.eu/archives/archiefweb/20180227121952/https://www.rijksoverheid.nl/binaries/rijksoverheid/documenten/rapporten/2012/08/13/black-tulip-update/black-tulip-update.pdf

Hallam-Baker, P., & Stradling, R. (2013, January). *RFC6844: DNS Certification Authority Authorization (CAA) Resource Record.* Retrieved from Internet Engineering Task Force: https://tools.ietf.org/pdf/rfc6844.pdf

Hein, B. (2013). The Threat Landscape of PKI: System and Cryptographic Security of X.509, Algorithms, and Their Implementations. *Proceedings of the Romanian Academy, Series A*, 286-294.

Internet Security Research Group. (2019, June 9). Retrieved from Let's Debug: https://letsdebug.net/

Langley, A. (2014, July 8). *Maintaining digital certificate security*. Retrieved from Google Security Blog: https://security.googleblog.com/2014/07/maintaining-digital-certificate-security.html

Microsoft. (2019, May 30). *Certificate Services Protocols Overview.* Retrieved from Open Specifications: https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-CERSOD/%5bMS-CERSOD%5d.pdf

O'Brien, D., Sleevi, R., & Whalley, A. (2017, September 11). *Chrome's Plan to Distrust Symantec Certificates*. Retrieved from Google Security Blog: https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html

Özarar, M. (2013, January 3). *Turktrust SubCA abuse and MITM'ing*. Retrieved from https://groups.google.com/forum/#!topic/mozilla.dev.security.policy/aqn0Zm-KxQ0%5B1-25%5D

Palmer, C. (2019, May 1). *Deprecations and removals in Chrome 67*. Retrieved from Google Developers: https://developers.google.com/web/updates/2018/04/chrome-67-deps-rems

Prins, J. R. (2011, September 5). *DigiNotar public report version 1.* Retrieved from Rijksoverheid: https://www.rijksoverheid.nl/documenten/rapporten/2011/09/05/diginotar-public-report-version-1

Smashing Magazine. (2016, October 26). *Be Afraid Of HTTP Public Key Pinning (HPKP)*. Retrieved from Smashing Magazine: https://www.smashingmagazine.com/be-afraid-of-public-key-pinning/

Stapleton, J. (2013, March). *PKI Under Attack.* Retrieved from ISSA Journal: https://cdn.ymaws.com/www.issa.org/resource/resmgr/JournalPDFs/PKI_Under_Attack_ISSA0313.pdf

Wakek, L., Mannan, M., & Youssef, A. (2018, September 24). *The Sorry State of TLS Security in Enterprise Interception Appliances*. Retrieved from Cornell University: https://arxiv.org/abs/1809.08729