

# Security

Geoff Huston  
Chief Scientist, APNIC



~~Security~~

insecurity!

Geoff Huston  
Chief Scientist, APNIC



# Which Bank?

The screenshot shows the CommBank website homepage. At the top, there is a navigation bar with the following links: BANKING, HOME BUYING, INSURANCE, INVESTING & SUPER, BUSINESS, and INSTITUTIONAL. A search icon, a refresh icon, and a 'Log on' button are also present in the navigation bar. The main content area features a large hero image of a woman in a yellow sweater standing by a window. Overlaid on this image is a white box with the text: 'You can count on CommBank in uncertain times.' Below this text, it says 'Continuing to serve our customers and support the economy.' and includes a button that says 'See coronavirus support'. Below the hero image, there is a horizontal menu with the following items: 'Explore products', 'Support', 'Rates & fees', and 'Tools & calculators'. The 'Explore products' item is highlighted with a yellow underline. Below this menu, there are four product tiles: 'Bank accounts' with a card icon, 'Credit cards' with a card icon, 'Savings with certainty' with a photo of a family and the text 'We're offering a range of Term', and 'Coronavirus support for home loan customers' with a photo of a woman on a laptop. A second 'Coronavirus support for business' tile is partially visible at the bottom right.

Personal banking including accounts, credit cards and home loans - CommBank

BANKING HOME BUYING INSURANCE INVESTING & SUPER BUSINESS INSTITUTIONAL

Log on

## You can count on CommBank in uncertain times.

Continuing to serve our customers and support the economy.

See coronavirus support

Explore products Support Rates & fees Tools & calculators

Bank accounts

Credit cards

Savings with certainty  
We're offering a range of Term

Coronavirus support for home loan customers

Coronavirus support for business

# Which Bank? My Bank!

The screenshot shows the CommBank website homepage. At the top, there is a navigation bar with a yellow diamond logo and links for BANKING, HOME BUYING, INSURANCE, INVESTING & SUPER, BUSINESS, and INSTITUTIONAL. A search icon, a refresh icon, and a 'Log on' button with a lock icon are also present. The main content area features a large hero section with a woman in a yellow sweater looking out a window. The headline reads 'You can count on CommBank in uncertain times.' Below this, it says 'Continuing to serve our customers and support the economy.' and includes a button that says 'See coronavirus support'. Underneath the hero section is a horizontal menu with links for 'Explore products', 'Support', 'Rates & fees', and 'Tools & calculators'. The bottom section contains four tiles: 'Bank accounts' with a card icon, 'Credit cards' with a card icon, 'Savings with certainty' with a photo of a family and the text 'We're offering a range of Term', and 'Coronavirus support for home loan customers' with a photo of a woman at a computer. Another tile for 'Coronavirus support for business' is partially visible on the right.

# Which Bank? My Bank!

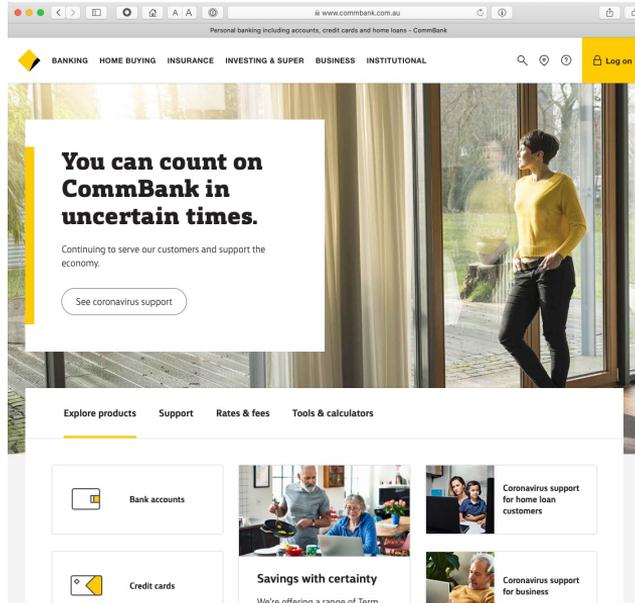
*i hope!*

The screenshot shows the CommBank website homepage. At the top, there is a navigation bar with links for BANKING, HOME BUYING, INSURANCE, INVESTING & SUPER, BUSINESS, and INSTITUTIONAL. A search icon, a refresh icon, and a 'Log on' button are also present. The main headline reads 'You can count on CommBank in uncertain times.' Below this, a sub-headline states 'Continuing to serve our customers and support the economy.' A button labeled 'See coronavirus support' is visible. A secondary navigation bar includes 'Explore products', 'Support', 'Rates & fees', and 'Tools & calculators'. The page features several content tiles: 'Bank accounts', 'Credit cards', 'Savings with certainty' (with a sub-note 'We're offering a range of Term'), 'Coronavirus support for home loan customers', and 'Coronavirus support for business'.

# Security on the Internet

How do you know that you are really going to where you thought you were going to?

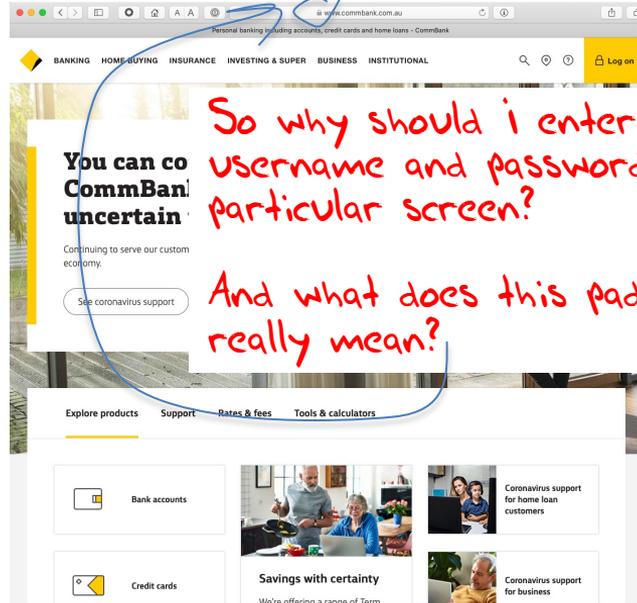
its trivial to mock up a web page to look like another



# Security on the Internet

How do you know that you are really going to where you thought you were going to?

its trivial to mock up a web page to look like another



So why should i enter my username and password into this particular screen?

And what does this padlock icon really mean?

# Opening the Connection: First Steps



Client:

*DNS Query:*

www.commbank.com.au?



*DNS Response:*

23.214.88.32

*TCP Session:*

TCP Connect 23.214.88.32, port 443



# Hang on...

```
$ dig -x 23.214.88.32 +short  
a23-214-88-32.deploy.static.akamaitechnologies.com.
```

# Hang on...

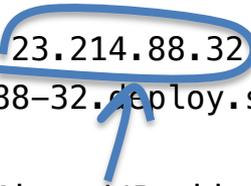
```
$ dig -x 23.214.88.32 +short  
a23-214-88-32.deploy.static.akamaitechnologies.com.
```

That's **not** an IP addresses that was allocated to the Commonwealth Bank!

The Commonwealth Bank of Australia has the address blocks  
140.168.0.0 - 140.168.255.255 and  
203.17.185.0 - 203.17.185.255

# Hang on...

```
$ dig -x 23.214.88.32 +short  
a23-214-88-32.deploy.static.akamaitechnologies.com.
```



That's an Akamai IP address

And I'm NOT a customer of the Internet Bank of Akamai!

Why should my browser trust that 23.214.88.32 is really the authentic web site for the Commonwealth Bank of Australia, and not some dastardly evil scam designed to steal my passwords and my money?

And why should I trust my browser?

# The major question...

How does my browser tell the difference between an intended truth and a dastardly lie?



# Public Key Cryptography

Pick a **pair** of keys such that:

- Messages encoded with one key can only be decoded with the other key
- Knowledge of the value of one key does not infer the value of the other key
- Make one key public, and keep the other a closely guarded private secret



# The Power of Primes

$$(m^e)^d \equiv m \pmod{n}$$

As long as  $d$  and  $n$  are relatively large, and  $n$  is the product of two large prime numbers, then finding the value of  $d$  when you already know the values of  $e$  and  $n$  is computationally expensive

# Why is this important?

Because much of the current foundation of Internet security rests upon this prime number relationship

Because prime number factorization still involves enumeration

And cryptography is still about getting the defender to perform just enough work to make the attacker's task so much greater that its infeasible

# Back to Public/Private Key Pairs

- If I have a copy of your PUBLIC key,
- And you encrypt a message with your PRIVATE key,
- Then I can decrypt the message.
  
- And I know it was you that sent it.
- And you can't deny it.

# Public Key Certificates

But how do I know this is YOUR public key?

– And not the public key of some dastardly evil agent pretending to be you?

- I don't know you
- I've never met you
- So I have absolutely no clue if this public key value is yours or not!

# Public Key Certificates

What if I 'trust' an intermediary?

- Who has contacted you and validated your identity and conducted a 'proof of possession' test that you have control of a private key that matches your public key
- Then if the intermediary signs an attestation that this is your public key (with their private key) then I would be able to trust this public key
- This 'attestation' takes the form of a "public key certificate"

# Public Key Certificates

- If the intermediary signs an attestation that this is a public key (with their private key) then
  - I trust this intermediary
  - And this intermediary has said that this is your public key
  - Then I can trust that this is your public key
- This ‘attestation’ takes the form of a “public key certificate”



Safari is using an encrypted connection to [www.commbank.com.au](https://www.commbank.com.au).

Encryption with a digital certificate keeps information private as it's sent to or from the https website [www.commbank.com.au](https://www.commbank.com.au).

DigiCert Inc has identified [www.commbank.com.au](https://www.commbank.com.au) as being owned by Commonwealth Bank of Australia in SYDNEY, New South Wales, AU.

DigiCert High Assurance EV Root CA  
↳ DigiCert SHA2 Extended Validation Server CA  
↳ [www.commbank.com.au](https://www.commbank.com.au)



[www.commbank.com.au](https://www.commbank.com.au)

Issued by: DigiCert SHA2 Extended Validation Server CA

Expires: Saturday, 23 July 2022 at 10:00:00 pm Australian Eastern Standard Time

✔ This certificate is valid

▶ Trust

▼ Details

Subject Name

**Business Category** Private Organization

**Inc. Country/Region** AU

**Serial Number** 123 123 124

**Country or Region** AU

**State/Province** New South Wales

**Locality** SYDNEY

**Organisation** Commonwealth Bank of Australia

**Organisational Unit** CBA Business System Hosting

**Common Name** www.commbank.com.au

Issuer Name

**Country or Region** US

**Organisation** DigiCert Inc

**Organisational Unit** www.digicert.com

**Common Name** DigiCert SHA2 Extended Validation Server CA

**Serial Number** 03 1A 62 D5 68 8B 27 9F 00 80 A9 D3 98 4F 41 66

**Version** 3

**Signature Algorithm** SHA-256 with RSA Encryption ( 1.2.840.113549.1.1.1 )

**Parameters** None

**Not Valid Before** Thursday, 25 June 2020 at 10:00:00 am Australian Eastern Standard Time

**Not Valid After** Saturday, 23 July 2022 at 10:00:00 pm Australian Eastern Standard Time

Public Key Info

**Algorithm** RSA Encryption ( 1.2.840.113549.1.1.1 )

**Parameters** None

**Public Key** 256 bytes : C5 48 B6 8B 2D 3F 67 3C ...

**Exponent** 65537

**Key Size** 2048 bits



Safari is using an encrypted connection to [www.potaroo.net](https://www.potaroo.net).

Encryption with a digital certificate keeps information private as it's sent to or from the https website [www.potaroo.net](https://www.potaroo.net).

DST Root CA X3  
Let's Encrypt Authority X3  
potaroo.net



### potaroo.net

Issued by: Let's Encrypt Authority X3

Expires: Saturday, 19 September 2020 at 12:43:45 pm Australian Eastern Standard Time

✔ This certificate is valid

▶ Trust

▼ Details

**Subject Name**  
**Common Name** potaroo.net

**Issuer Name**  
**Country or Region** US  
**Organisation** Let's Encrypt  
**Common Name** Let's Encrypt Authority X3

**Serial Number** 04 D0 18 DF ED 7A F7 3F 8D DA 29 45 76 64 A1 E9 E9 04  
**Version** 3

**Signature Algorithm** SHA-256 with RSA Encryption (1.2.840.113549.1.1.1)  
**Signature** 256 bytes : 65 94 89 F8 3A 04 C2 53 ...

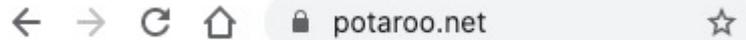
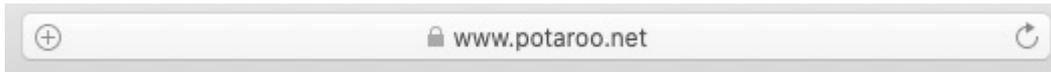
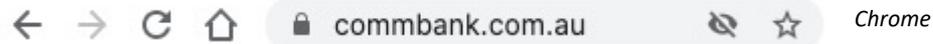
**Public Key** 256 bytes : BF 24 1A 56 39 86 01 30 ...  
**Exponent** 65537  
**Key Size** 2,048 bits  
**Key Usage** Encrypt, Verify, Wrap, Derive

**Signature** 256 bytes : 65 94 89 F8 3A 04 C2 53 ...

**Extension** Key Usage ( 2.5.29.15 )  
**Critical** YES  
**Usage** Digital Signature, Key Encipherment

*Not all certificates are the same - This certificate binds a public key to a set of names without any attestation to the identity of the name "holder"*

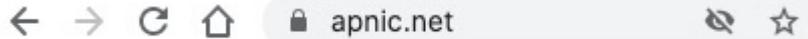
# Spot the Difference



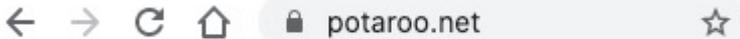
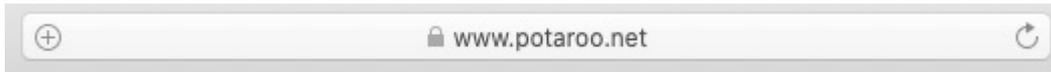
# Spot the Difference



This web site's certificate was issued to an organisation called the "Commonwealth Bank of Australia" located in Sydney, Australia



This web site's certificate was issued to "Cloudflare Inc" located in San Francisco, USA!!

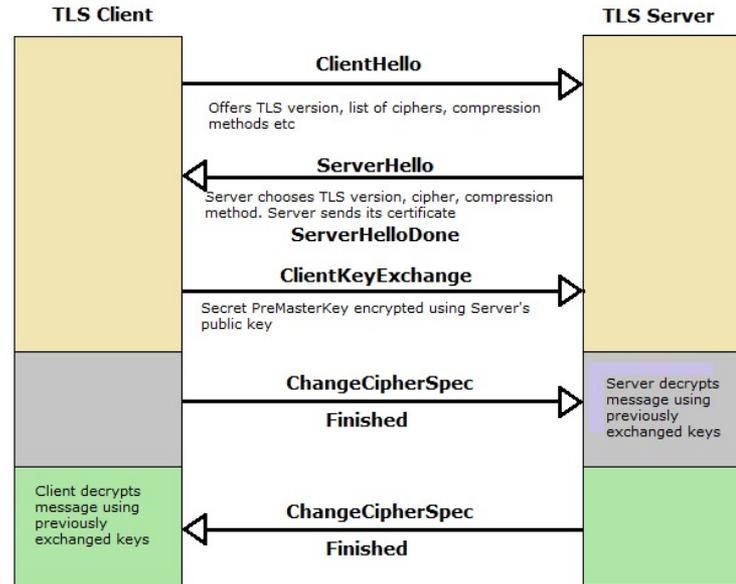


This web site's certificate says *nothing* about the entity that holds the public key associated with this domain

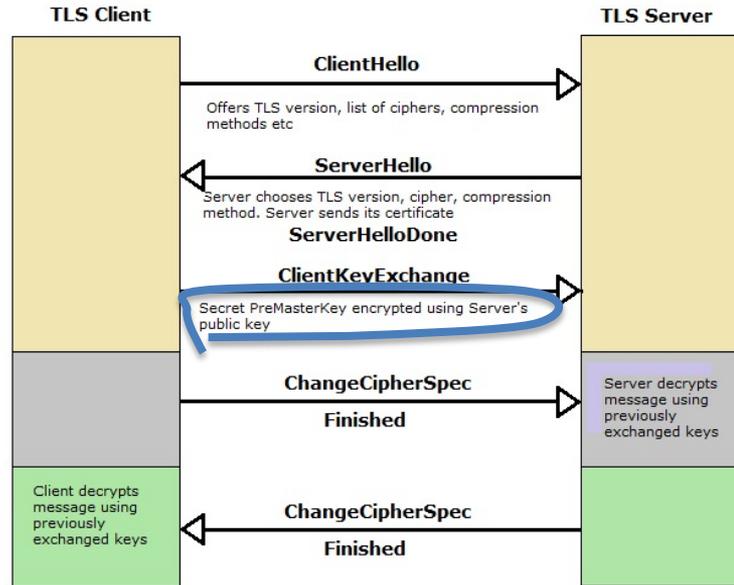
## Moving on...

- Ok, so the certificate system is a total mess, but TLS still works, right?

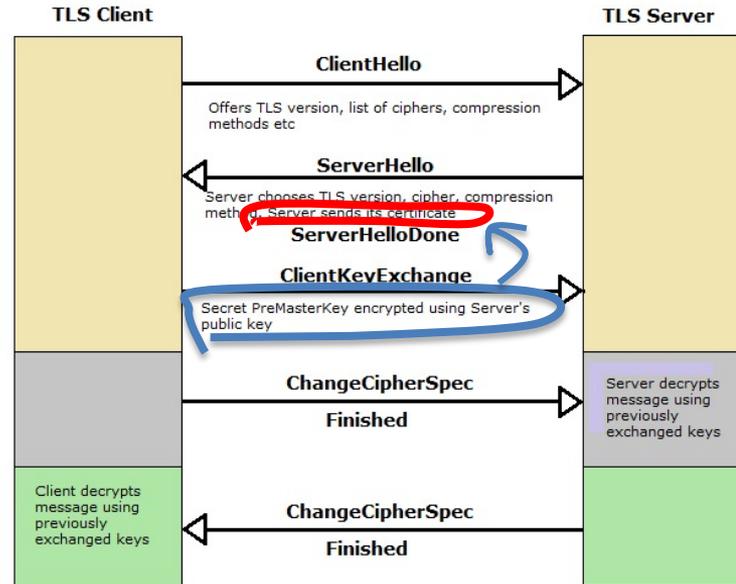
# Secure Connections using TLS



# Secure Connections using TLS



# Secure Connections using TLS





Safari is using an encrypted connection to [www.commbank.com.au](https://www.commbank.com.au).

Encryption with a digital certificate keeps information private as it's sent to or from the https website [www.commbank.com.au](https://www.commbank.com.au).

DigiCert Inc has identified [www.commbank.com.au](https://www.commbank.com.au) as being owned by Commonwealth Bank of Australia in SYDNEY, New South Wales, AU.

DigiCert High Assurance EV Root CA  
DigiCert SHA2 Extended Validation Server CA  
[www.commbank.com.au](https://www.commbank.com.au)

 **www.commbank.com.au**  
Issued by: DigiCert SHA2 Extended Validation Server CA  
Expires: Thursday, 23 July 2020 at 10:00:00 pm Australian Eastern Standard Time  
✔ This certificate is valid

Trust  
Details

**Subject Name** \_\_\_\_\_  
**Business Category** Private Organization  
**Inc. Country/Region** AU  
**Serial Number** 123 123 124  
**Country or Region** AU  
**State/Province** New South Wales  
**Locality** SYDNEY  
**Organisation** Commonwealth Bank of Australia  
**Organisational Unit** CBA Business System Hosting  
**Common Name** [www.commbank.com.au](https://www.commbank.com.au)

**Issuer Name** \_\_\_\_\_  
**Country or Region** US  
**Organisation** DigiCert Inc  
**Organisational Unit** [www.digicert.com](https://www.digicert.com)  
**Common Name** DigiCert SHA2 Extended Validation Server CA

**Serial Number** 06 E0 31 4E 58 49 08 51 FC 66 31 FD 7D 21 D9 24  
**Version** 3  
**Signature Algorithm** SHA-256 with RSA Encryption ( 1.2.840.113549.1.1.1 )  
**Parameters** None

**Not Valid Before** Thursday, 16 May 2019 at 10:00:00 am Australian Eastern Standard Time  
**Not Valid After** Thursday, 23 July 2020 at 10:00:00 pm Australian Eastern Standard Time

**Public Key Info**

**Algorithm** RSA Encryption ( 1.2.840.113549.1.1.1 )  
**Parameters** None  
**Public Key** 256 bytes : AF 2C 6F 6B 07 E6 54 C0 ...  
**Exponent** 65537  
**Key Size** 2,048 bits  
**Key Usage** Encrypt, Verify, Wrap, Derive  
**Signature** 256 bytes : C6 15 E5 ED 54 85 84 67 ...

**Extension** Key Usage ( 2.5.29.15 )  
**Critical** YES  
**Usage** Digital Signature, Key Encipherment

**Extension** Basic Constraints ( 2.5.29.19 )  
**Critical** NO

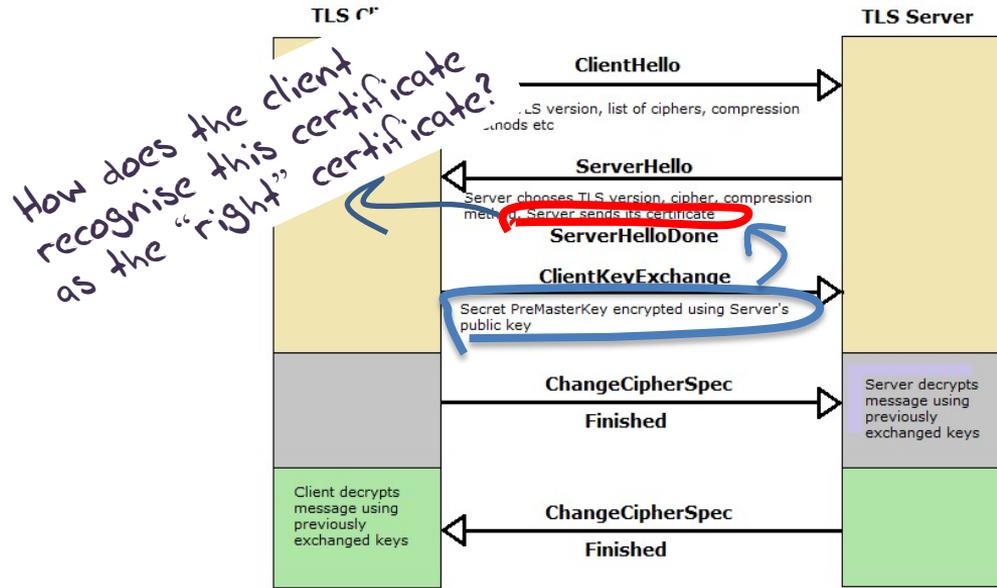
**Certificate Authority** NO



Hide Certificate

OK

# Secure Connections using TLS





Safari is using an encrypted connection to [www.commbank.com.au](https://www.commbank.com.au).

Encryption with a digital certificate keeps information private as it's sent to or from the https website [www.commbank.com.au](https://www.commbank.com.au).

DigiCert Inc has identified [www.commbank.com.au](https://www.commbank.com.au) as being owned by Commonwealth Bank of Australia in SYDNEY, New South Wales, AU.

DigiCert High Assurance EV Root CA  
DigiCert SHA2 Extended Validation Server CA  
[www.commbank.com.au](https://www.commbank.com.au)

 **www.commbank.com.au**  
Issued by DigiCert SHA2 Extended Validation Server CA  
Expires: Thursday, 23 July 2020 at 10:00:00 pm Australian Eastern Standard Time  
✔ This certificate is valid

Trust  
Details

**Subject Name**

Business Category	Private Organization
Inc. Country/Region	AU
Serial Number	123 123 124
Country or Region	AU
State/Province	New South Wales
Locality	SYDNEY
Organisation	Commonwealth Bank of Australia
Organisational Unit	CBA Business System Hosting
Common Name	<a href="https://www.commbank.com.au">www.commbank.com.au</a>

**Issuer Name**

Country or Region	US
Organisation	DigiCert Inc
Organisational Unit	<a href="https://www.digicert.com">www.digicert.com</a>
Common Name	DigiCert SHA2 Extended Validation Server CA

**Serial Number** 06 E0 31 4E 58 49 08 51 FC 66 31 FD 7D 21 D9 24  
**Version** 3  
**Signature Algorithm** SHA-256 with RSA Encryption ( 1.2.840.113549.1.1.11 )  
**Parameters** None

**Not Valid Before** Thursday, 16 May 2019 at 10:00:00 am Australian Eastern Standard Time  
**Not Valid After** Thursday, 23 July 2020 at 10:00:00 pm Australian Eastern Standard Time

**Public Key Info**

Algorithm	RSA Encryption ( 1.2.840.113549.1.1.1 )
Parameters	None
Public Key	256 bytes : AF 2C 6F 6B 07 E6 54 C0 ...
Exponent	65537
Key Size	2,048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : C6 15 E5 ED 54 85 84 67 ...

**Extension** Key Usage ( 2.5.29.15 )  
**Critical** YES  
**Usage** Digital Signature, Key Encipherment

**Extension** Basic Constraints ( 2.5.29.19 )  
**Critical** NO

**Certificate Authority** NO

? How did my browser know that this is a "valid" cert?

Hide Certificate

OK

# Domain Name Certification

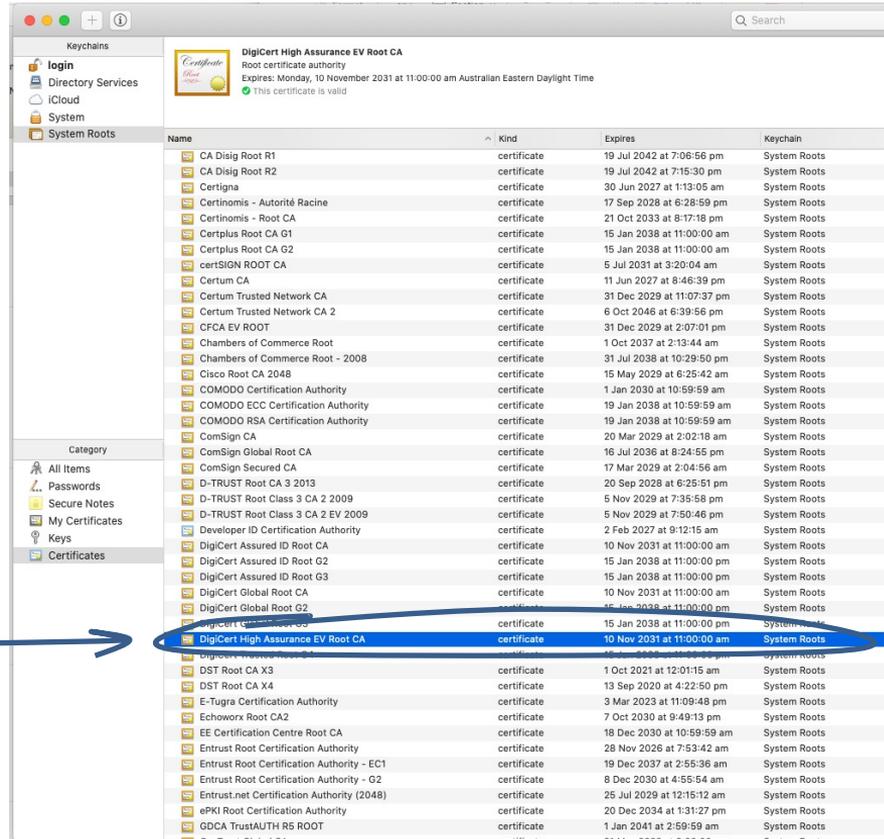
- The Commonwealth Bank of Australia has generated a key pair
- And they passed a certificate signing request to a company called “DigiCert Inc” in the US
- Who was willing to vouch (in a certificate) that the entity is called the Commonwealth Bank of Australia and they have control of the the domain name [www.commbank.com.au](http://www.commbank.com.au) and they have a certain public key
- So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to an entity that is able to demonstrate knowledge of the private key for [www.commbank.com.au](http://www.commbank.com.au), as long as I am prepared to trust DigiCert and the certificates that they issue
- And I’m prepared to trust them because DigiCert NEVER lie!

# Domain Name Certification

- The Commonwealth Bank of Australia has generated a key pair
- And they passed a certificate signing request to a company called “DigiCert Inc” in the US
- Who was willing to vouch (in a certificate) that the entity is called the Commonwealth Bank of Australia and they have control of the the domain name [www.commbank.com.au](http://www.commbank.com.au) and they have a certain public key
- So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to an entity that is able to demonstrate knowledge of the private key for [www.commbank.com.au](http://www.commbank.com.au), as long as I am prepared to trust DigiCert and the certificates that they issue
- And I’m *How do i know that? Why should i trust them?* because DigiCert NEVER lie!

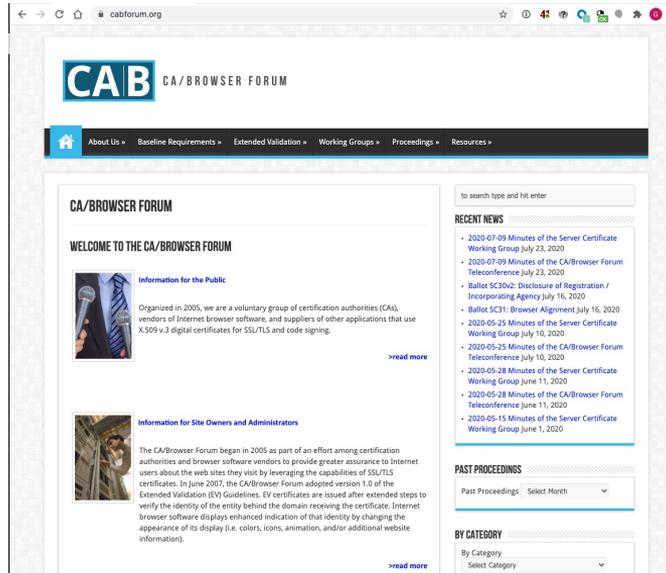
# Local Trust

The cert i'm being asked to trust was issued by a certification authority that my browser already trusts - so i trust that cert!



# Local Trust

These Certificate Authorities are listed in my computer's trust set because they claim to operate according to the practices defined by the CAB industry forum (of which they are a member) and they **never** lie!



The screenshot shows the homepage of the CAB/BROWSER FORUM website. The browser address bar displays 'cabforum.org'. The website header features the CAB logo and the text 'CA/BROWSER FORUM'. A navigation menu includes links for 'About Us', 'Baseline Requirements', 'Extended Validation', 'Working Groups', 'Proceedings', and 'Resources'. The main content area is titled 'CA/BROWSER FORUM' and 'WELCOME TO THE CA/BROWSER FORUM'. It contains two sections: 'Information for the Public' and 'Information for Site Owners and Administrators'. The 'Information for the Public' section includes a paragraph about the forum's purpose and a 'read more' link. The 'Information for Site Owners and Administrators' section includes a paragraph about the forum's history and a 'read more' link. On the right side, there are two sections: 'RECENT NEWS' and 'PAST PROCEEDINGS'. The 'RECENT NEWS' section lists several recent events, including '2020-07-09 Minutes of the Server Certificate Working Group July 23, 2020' and '2020-07-09 Minutes of the CA/Browser Forum Teleconference July 23, 2020'. The 'PAST PROCEEDINGS' section has a dropdown menu for 'Past Proceedings: Select Month'. At the bottom right, there is a 'BY CATEGORY' section with a dropdown menu for 'By Category: Select Category'.

# Local Trust

These Certificate Authorities are listed in my computer's trust set because they claim to operate according to the practices defined by the CAB industry forum (of which they are a member) and they **never** lie!

*So somebody (i have never met) paid someone else (whom i have also never met) some money and then my browser trusts everything they have ever done and everything they will ever do in the future - ok?*

A screenshot of the CAB website content area. It includes a section titled 'Information for Site Owners and Administrators' with a small image of a person and a paragraph of text. Below this is a 'PAST PROCEEDINGS' section with a dropdown menu for 'Select Month'. At the bottom is a 'BY CATEGORY' section with a dropdown menu for 'Select Category'. There are also 'read more' links on the right side of the text blocks.

**Information for Site Owners and Administrators**

The CA/Browser Forum began in 2005 as part of an effort among certification authorities and browser software vendors to provide greater assurance to Internet users about the web sites they visit by leveraging the capabilities of SSL/TLS certificates. In June 2007, the CA/Browser Forum adopted version 1.3 of the Extended Validation (EV) Guidelines. EV certificates are issued after extended steps to verify the identity of the entity behind the domain receiving the certificate. Internet browser software displays enhanced indication of that identity by changing the appearance of its display (i.e. colors, icons, animation, and/or additional website information).

**PAST PROCEEDINGS**

Past Proceedings:

**BY CATEGORY**

By Category:

# Local Trust or Local Credulity\*?

Wow!

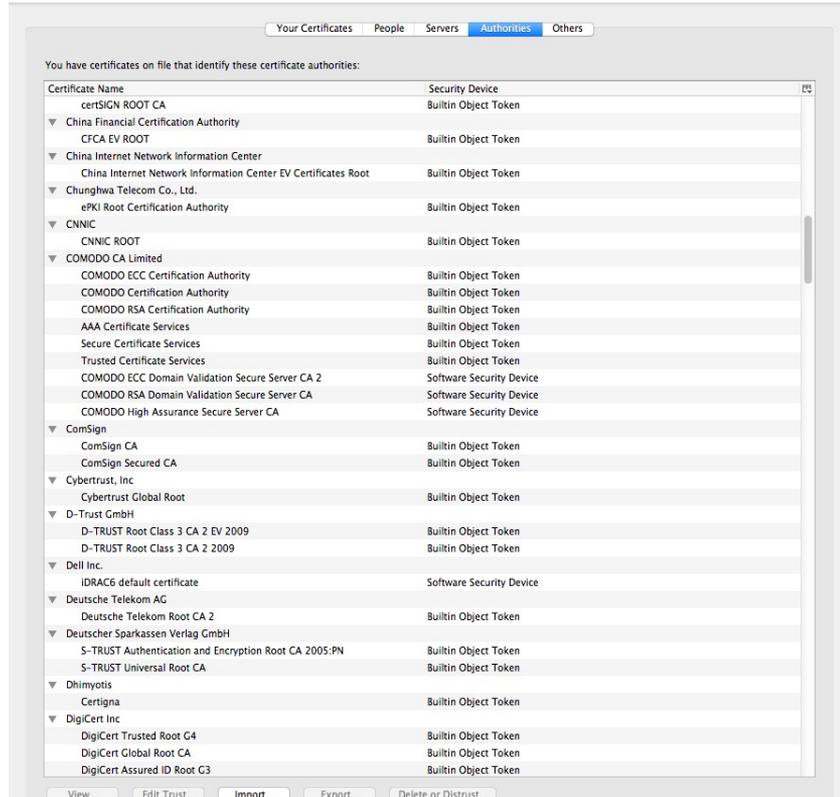
Are they **all** trustable?

\* cre·du·li·ty

/kraˈd(y)oələdē/

noun

a tendency to be too ready to believe that something is real or true.



# Local Credulity

Wow!

Are they all trustable?

*Evidently Not!*

The image shows a Windows Certificate Manager window with the 'Authorities' tab selected. The list of certificate authorities includes:

- certSIGN ROOT CA
- China Financial Certification Authority
- CFCA EV ROOT
- China Internet Network Information Center
- China Internet Network Information Center EV Certificates Root
- Chungghwa Telecom
- CNNIC** (highlighted with a blue circle)
- CNNIC ROOT
- COMODO CA Limited
- COMODO ECC C
- COMODO Certif
- COMODO RSA C
- AAA Certificate
- Secure Certifica
- Trusted Certific
- COMODO ECC C
- COMODO RSA C
- COMODO High
- ComSign
- ComSign CA
- ComSign Secur
- Cybertrust, Inc
- Cybertrust Glob
- D-Trust GmbH
- D-TRUST Root C
- D-TRUST Root F
- Dell Inc.
- IDRAC6 default
- Deutsche Telekom
- Deutsche Telek
- Deutscher Sparkas
- S-TRUST Auther
- S-TRUST Univer
- Dhimyotis
- Certigna
- DigiCert Inc
- DigiCert Trustee
- DigiCert Global
- DigiCert Assure

The right side of the image shows a blog post titled "Maintaining digital certificate security" by Adam Langley, Security Engineer, dated Monday, March 23, 2015. A blue circle highlights a paragraph in the post:

CNNIC is included in all major root stores and so the misissued certificates would be trusted by almost all browsers and operating systems. Chrome on Windows, OS X, and Linux, ChromeOS, and Firefox 33 and greater would have rejected these certificates because of **public-key pinning**, although misissued certificates for other sites likely exist.

# Local Credulity

Wow!

Are they all trustable?

*Evidently Not!*

The screenshot shows a web browser window with two tabs. The active tab is titled "The real security issue behind the Comodo hack | InfoWorld" and displays an article by Roger A. Grimes. The article title is "The real security issue behind the Comodo hack" with a subtitle "The Comodo hack has grabbed headlines, but more troubling is the public's ignorance over PKI and digital certificates". A blue circle highlights a paragraph in the article: "News of an Iranian hacker duping certification authority Comodo into issuing digital certificates to one or more unauthorized parties has caused an uproar in the IT community, moving some critics to call for Microsoft and Mozilla to remove Comodo as a trusted root certification authority from the systems under their control. Though the hacker managed to never first compromising a site containing a hard-coded logon name and password, then generating certificates for several well-known sites, including Google, Live.com, Skype, and Yahoo, I'm not bothered by the".

On the left side of the browser window, a list of certificate authorities is visible, including:

- certSIGN ROOT CA
- China Financial Certification Authority
- CFCA EV ROOT
- China Internet Network Informatic
- China Internet Network Inform
- Chunghwa Telecom Co., Ltd.
- ePKI Root Certification Authority
- CNNIC
- CNNIC ROOT
- COMODO CA Limited
- COMODO ECC Certification Authority
- COMODO Certification Authority
- COMODO RSA Certification Authority
- AAA Certificate Services
- Secure Certificate Services
- Trusted Certificate Services
- COMODO ECC Domain Validation
- COMODO RSA Domain Validation
- COMODO High Assurance Secu
- ComSign
- ComSign CA
- ComSign Secured CA
- Cybertrust, Inc
- Cybertrust Global Root
- D-Trust GmbH
- D-TRUST Root Class 3 CA 2 EV
- D-TRUST Root Class 3 CA 2 TC
- Dell Inc.
- IDRAC6 default certificate
- Deutsche Telekom AG
- Deutsche Telekom Root CA 2
- Deutscher Sparkassen Verlag Gm
- S-TRUST Authentication and E
- S-TRUST Universal Root CA
- Dhimyotis
- Certigna
- DigiCert Inc
- DigiCert Trusted Root G4
- DigiCert Global Root CA
- DigiCert Assured ID Root G3

Never?

# Well, hardly ever

ars TECHNICA **BIZ & IT** TECH SCIENCE POLICY CARS GAMING & CULTURE FORUMS

RISK ASSESSMENT —

## Already on probation, Symantec issues more illegit HTTPS certificates

At least 108 Symantec certificates threatened the integrity of the encrypted Web.

DAN GOODIN • 1/21/2017, 8:40 AM



Enlarge

62

A security researcher has unearthed evidence showing that three browser-trusted certificate authorities (CAs) owned and operated by Symantec improperly issued more than 100 unvalidated [transport layer security](#) certificates. In some cases, those certificates made it possible to spoof HTTPS-protected websites.

<http://arstechnica.com/security/2017/01/already-on-probation-symantec-issues-more-illegit-https-certificates/>

### Misissued/Suspicious Symantec Certificates

Andrew Ayer | Thu, 19 Jan 2017 13:47:06 -0800

I. Misissued certificates for example.com

On 2016-07-14, Symantec misissued the following certificates for example.com:

<https://crt.sh/?sha256=A8F14F52CC1282D7153A13316E7DA39E6AE37B1A10C16288B9024A9B9DC3C4C6>

<https://crt.sh/?sha256=8B5956C57FDC720B6907A4B1BC8CA2E46CD90EAD5C061A426CF48A6117BF8FA>

<https://crt.sh/?sha256=94482136A1400BC3A1136FPCA3E79D4D200E03DD20B245D19F0E78B5679EA48>

<https://crt.sh/?sha256=C69A804C1B20E6FC7861C67476CADD1DAE7A8DCF6E23E15311C2D2794BFCDD11>

I confirmed with ICANN, the owner of example.com, that they did not authorize these certificates. These certificates were already revoked at the time I found them.

II. Suspicious certificates for domains containing the word "test"

On 2016-11-15 and 2016-10-26, Symantec issued certificates for various domains containing the word "test" which I strongly suspect were misissued:

# Well, hardly ever



## Google Security Blog

The latest news and insights from Google on security and safety on the Internet

### Distrust of the Symantec PKI: Immediate action needed by site operators

March 7, 2018

Posted by Devon O'Brien, Ryan Sleevi, Emily Stark, Chrome security team

We [previously announced](#) plans to deprecate Chrome's trust in the Symantec certificate authority (including Symantec-owned brands like Thawte, VeriSign, Equifax, GeoTrust, and RapidSSL). This post outlines how site operators can determine if they're affected by this deprecation, and if so, what needs to be done and by when. Failure to replace these certificates will result in site breakage in upcoming versions of major browsers, including Chrome.

#### Chrome 66

If your site is using a SSL/TLS certificate from Symantec that was issued before June 1, 2016, it will stop functioning in Chrome 66, which could already be impacting your users.

If you are uncertain about whether your site is using such a certificate, you can preview these changes in [Chrome Canary](#) to see if your site is affected. If connecting to your site displays a certificate error or a warning in DevTools as shown below, you'll need to replace your certificate. You can get a new certificate from any [trusted CA](#), including Digicert, which recently acquired Symantec's CA business.

With unpleasant consequences when it all  
goes wrong



BORDER GATEWAY PROTOCOL ATTACK —

# Suspicious event hijacks Amazon traffic for 2 hours, steals cryptocurrency

Almost 1,300 addresses for Amazon Route 53 rerouted for two hours.

DAN GOODIN - 4/25/2018, 5:00 AM

The image shows the Amazon.com logo, consisting of the word "amazon.com" in a bold, black, sans-serif font. Below the text is the iconic orange arrow that curves from the letter 'a' to the letter 'z', representing a smile.

Amazon

123



Amazon lost control of a small number of its cloud services IP addresses for two hours on Tuesday morning when hackers exploited a known Internet-protocol weakness that let them to redirect traffic to rogue destinations. By subverting Amazon's domain-resolution service, the attackers masqueraded as cryptocurrency website MyEtherWallet.com and stole about \$150,000 in digital coins from unwitting end users. They may have targeted other Amazon customers as well.

The incident, which started around 6 AM California time, hijacked roughly 1,300 IP addresses, Oracle-owned Internet Intelligence [said on Twitter](#). The malicious redirection was caused by fraudulent routes that were announced by [Columbus, Ohio-based eNet](#), a large Internet service provider that is referred to as autonomous system 10297. Once in place, the eNet announcement caused Hurricane Electric and possibly Hurricane Electric customers and other eNet peers to send traffic over the same unauthorized routes. The 1,300 addresses belonged to [Route 53](#), Amazon's domain name system service

The attackers managed to steal about \$150,000 of currency from MyEtherWallet users,

What's going wrong here?

# What's going wrong here?

- The TLS handshake cannot specify WHICH CA should be used by the client to validate the digital certificate that describes the server's public key
- The result is that your browser will allow ANY CA to be used to validate a certificate!

# What's going wrong here?

- The TLS handshake cannot specify WHICH CA should be used by the client to validate the digital certificate that describes the server's public key
- The result is that your browser will allow ANY CA to be used to validate a certificate!

*WOW! That's awesomely bad!*

# What's going wrong here?

- The TLS handshake cannot specify WHICH CA



Here's a lock - it might be the lock on your front door for all I know.

The lock might LOOK secure, but don't worry - literally ANY key can open it!

validate a certificate!

WOW! That's awesomely bad!

sh  
dig  
pu  
• Th  
CA

!  
S  
NY

# What's going wrong here?

- There is no incentive for quality in the CA marketplace
- Why pay more for any certificate when the entire CA structure is only as strong as the weakest CA
- And your browser trusts a LOT of CAs!
  - About 60 – 100 CA's
  - About 1,500 Subordinate RA's
  - Operated by 650 different organisations

*See the EFF SSL observatory*

*<http://www.eff.org/files/DefconSSLiverse.pdf>*

# In a commercial environment

Where CA's compete with each other for market share

And quality offers no protection

Then what 'wins' in the market?



?

# In a commercial environment

Where CA's compete with each other for market share

And quality offers no protection

Then what 'wins' in the market?

Sustainable  
Resilient

Secure

Privacy

Trusted



cheap!

But its all OK

Really.

- Because 'bad' certificates can be revoked
- And browsers **always** check revocation status of certificates

Always?

Ok - Not Always.  
Some do.  
Sometimes.

Platform	Chrome	Firefox	Opera	Safari	Edge
Mac OS X 10.15.3	YES 80.0.3987.132	YES 73.0.1	YES 67.0.3575.53	YES 13.0.5	
iOS 13.3.1	YES 80.0.3987.95	YES 23.0	NO 16.0.15	YES 13.3.1	
Android 10	NO 80.0.3987.132	NO 68.6.0	NO 56.1		
Windows 10	NO 80.0.3987.132	YES 74.0	NO 67		YES 44.18362

Table 1 – Browser Revocation Status

# How can we fix this?

Option A: Take all the money out of the system!



The image shows a screenshot of the Let's Encrypt website. At the top left is the Let's Encrypt logo, which consists of a padlock with a sunburst above it, followed by the text "Let's Encrypt". To the right of the logo is the text "LINUX FOUNDATION COLLABORATIVE PROJECTS". Below this, there is a navigation menu with the following items: "Documentation", "Get Help", "Donate", and "About Us". The main content area features a large, semi-transparent white box with a geometric, low-poly background. Inside this box, the text reads: "Let's Encrypt is a **free, automated, and open** Certificate Authority." Below this text are two buttons: "Get Started" and "Donate".

# How can we fix this?

Option A: Take all the money out of the system!



The image shows a screenshot of the Let's Encrypt website. At the top left is the Let's Encrypt logo, and at the top right is the text "LINUX FOUNDATION COLLABORATIVE PROJECTS". Below the logo are navigation links: "Documentation", "Get Help", "Donate", and "About Us". The main content area features a large banner with a geometric background. Overlaid on this banner is handwritten text in brown ink that reads: "Will the automation of the Cert issuance coupled with a totally free service make the overall environment more or less secure?" Below this text are two buttons labeled "Get Started" and "Donate". At the bottom of the handwritten text, it says "i think we already know the answer!".

Let's Encrypt is a free, automated, and open Certificate Authority.

Get Started Donate

i think we already know the answer!

# How can we fix this?

## Option B: White Listing and Pinning with HSTS

[https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport\\_security\\_state\\_static.json](https://code.google.com/p/chromium/codesearch#chromium/src/net/http/transport_security_state_static.json)

```
transport_security_state_static.json
1 // Copyright (c) 2012 The Chromium Authors. All rights reserved.
2 // Use of this source code is governed by a BSD-style license that can be
3 // found in the LICENSE file.
4
5 // This file contains the HSTS preloaded list in a machine readable format.
6
7 // The top-level element is a dictionary with two keys: "pinsets" maps details
8 // of certificate pinning to a name and "entries" contains the HSTS details for
9 // each host.
10 //
11 // "pinsets" is a list of objects. Each object has the following members:
12 //   name: (string) the name of the pinset
13 //   static_spki_hashes: (list of strings) the set of allowed SPKIs hashes
14 //   bad_static_spki_hashes: (optional list of strings) the set of forbidden
15 //     SPKIs hashes
16 //   report_uri: (optional string) the URI to send violation reports to;
17 //     reports will be in the format defined in RFC 7469
18 //
19 // For a given pinset, a certificate is accepted if at least one of the
20 // "static_spki_hashes" SPKIs is found in the chain and none of the
21 // "bad_static_spki_hashes" SPKIs are. SPKIs are specified as names, which must
22 // match up with the file of certificates.
23 //
```

# How can we fix this?

## Option B: White Listing and Pinning with HSTS

[https://code.google.com/p/chromium/source/trunk/net/http/transport\\_security\\_state\\_static.json](https://code.google.com/p/chromium/source/trunk/net/http/transport_security_state_static.json) *its not a totally insane idea -- until you realise that it appears to be completely unscalable!*

*its just Google protecting itself and no one else*

```
transport_security_state_static.json
Copyright 2014 Google Inc. All Rights reserved.
of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// This file contains the HSTS preloaded list in a machine readable format.
6
7 // The top-level element is a dictionary with two keys: "pinsets" maps details
8 // of certificate pinning to a name and "entries" contains the HSTS details for
9 // each host.
10 //
11 // "pinsets" is a list of objects. Each object has the following members:
12 //   name: (string) the name of the pinset
13 //   static_spki_hashes: (list of strings) the set of allowed SPKIs hashes
14 //   bad_static_spki_hashes: (optional list of strings) the set of forbidden
15 //     SPKIs hashes
16 //   report_uri: (optional string) the URI to send violation reports to;
17 //     reports will be in the format defined in RFC 7469
18 //
19 // For a given pinset, a certificate is accepted if at least one of the
20 // "static_spki_hashes" SPKIs is found in the chain and none of the
21 // "bad_static_spki_hashes" SPKIs are. SPKIs are specified as names, which must
22 // match up with the file of certificates.
23 //
```

# How can we fix this?

O: its not a totally insane idea -- until you realise that it appears to be completely unscalable!

<http://code.google.com/p/chromium/codesearch#chromium/src/net/http/tree/security/static/icon>  
else



**INFOWORLD TECH WATCH**

By Fahmida Y. Rashid, Senior Writer, InfoWorld | JAN 30, 2017

About |

Informed news analysis every weekday

## Google moves into the Certificate Authority business

Google doesn't seem to trust the current system, as it has launched its own security certificates

```
17 // reports will be in the format defined in RFC 7469
18 //
19 // For a given pinset, a certificate is accepted if at least one of the
20 // "static_spki_hashes" SPKIs is found in the chain and none of the
21 // "bad_static_spki_hashes" SPKIs are. SPKIs are specified as names, which must
22 // match up with the file of certificates.
23 //
```

# How can we fix this?

## Option C: Certificate Transparency

Google Transparency Report

Overview Certificates

### HTTPS encryption on the web

#### Certificate transparency

In order to provide encrypted traffic to users, a site must first apply for a certificate from a trusted Certificate Authority (CA). This certificate is then presented to the browser to authenticate the site the user is trying to access. In recent years, due to structural flaws in the HTTPS certificate system, certificates and issuing CAs have proven vulnerable to compromise and manipulation. Google's Certificate Transparency project aims to safeguard the certificate issuance process by providing an open framework for monitoring and auditing HTTPS certificates.

Use the search bar below to look up all of a domain's certificates that are present in [active public certificate transparency logs](#). Site owners can search this site for domain names they control to ensure there have been no incorrect issuances of certificates referencing their domains.

Google encourages all CAs to write the certificates they issue to publicly verifiable, append-only, tamper-proof logs. In the future, Chrome and other browsers may decide not to accept certificates that have not been written to such logs.

As of May 6, 2020, there have been 9,178,649,266 entries made to the set of Certificate Transparency logs that Google monitors.

[Learn more about the Certificate Transparency Project](#)

#### Search certificates by hostname

www.potaroo.net

include subdomains

Current status:

Issuer	# Issued
C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3	36 <a href="#">Filter</a>

Subject	Issuer	# DNS names	Valid from	Valid to	# CT logs
*.potaroo.net	Let's Encrypt Authority X3	1	Mar 29, 2020	Jun 27, 2020	4 <a href="#">See details</a>
www.potaroo.net	Let's Encrypt Authority X3	1	Oct 21, 2019	Jan 19, 2020	4 <a href="#">See details</a>
www.potaroo.net	Let's Encrypt Authority X3	1	Aug 22, 2019	Nov 20, 2019	6 <a href="#">See details</a>

# How can we fix this?

## Option C: Certificate Transparency

The screenshot shows the Google Transparency Report interface. At the top, it says "Google Transparency Report" with a hamburger menu icon. Below that are tabs for "Overview" and "Certificates". The main heading is "HTTPS encryption on the web". Underneath, there's a section titled "Certificate transparency" with a sub-heading "Certificate transparency". The text explains that in order to provide encrypted traffic, a site must first apply for a certificate from a trusted Certificate Authority (CA). It mentions that in recent years, due to structural flaws in the HTTPS certificate system, certificates and issuing CAs have proven vulnerable to compromise and manipulation. Google's Certificate Transparency project aims to safeguard the certificate issuance process by providing an open framework for monitoring and auditing HTTPS certificates. A search bar is present, and a note states that as of May 18, 2019, there have been 9,178,649,266 entries made to the set of Certificate Transparency logs that Google monitors. A link "Learn more about the Certificate Transparency Project" is provided.

This is true

In order to provide encrypted traffic to users, a site must first apply for a certificate from a trusted Certificate Authority (CA). This certificate is then presented to the browser to authenticate the site the user is trying to access. In recent years, due to structural flaws in the HTTPS certificate system, certificates and issuing CAs have proven vulnerable to compromise and manipulation. Google's Certificate Transparency project aims to safeguard the certificate issuance process by providing an open framework for monitoring and auditing HTTPS certificates.

Current status:

Issuer	# Issued
C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3	36 <a href="#">Filter</a>

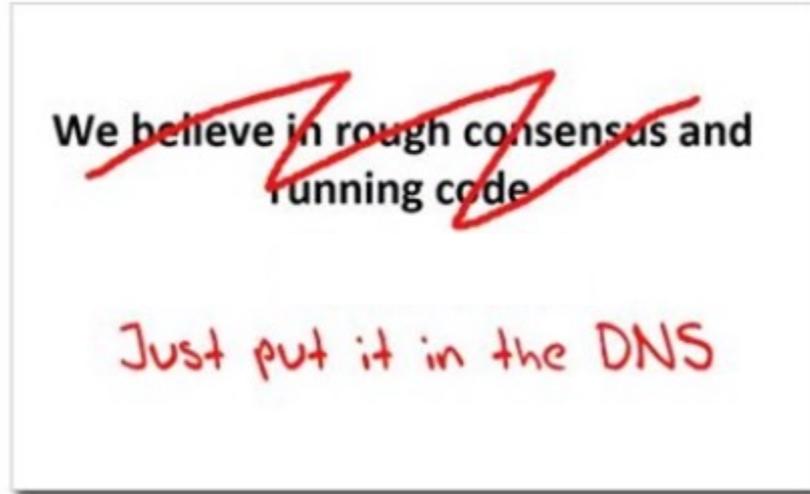
Subject	Issuer	# DNS names	Valid from	Valid to	# CT logs
*.potaroo.net	Let's Encrypt Authority X3	1	Mar 29, 2020	Jun 27, 2020	4 <a href="#">See details</a>
www.potaroo.net	Let's Encrypt Authority X3	1	Oct 21, 2019	Jan 19, 2020	4 <a href="#">See details</a>
www.potaroo.net	Let's Encrypt Authority X3	1	Aug 22, 2019	Nov 20, 2019	6 <a href="#">See details</a>

This is a fail



# How can we fix this?

Option D: Use the DNS!



# Seriously? The DNS?

Where better to find out the public key associated with a DNS-named service than to look it up in the DNS?

- Why not query the DNS for the HSTS record?
- Why not query the DNS for the issuer CA?
- Why not query the DNS for the hash of the domain name cert?
- Why not query the DNS for the hash of the domain name public key?

# Seriously? The DNS?

Where better to find out the public key associated with a DNS-named service than to look it up in the DNS?

- Why not query the DNS for the HSTS
- Why not query the DNS for the CA?
- Why not query the DNS for the hash of the domain name cert?
- Why not query the DNS for the hash of the domain name public key?

*Who needs CA's anyway?*

# DANE

- Using the DNS to associated domain name public key certificates with domain name

[\[Docs\]](#) [\[txt\]](#) [\[pdf\]](#) [\[draft-ietf-dane-p...\]](#) [\[Diff1\]](#) [\[Diff2\]](#) [\[Errata\]](#)

Updated by: [7218](#), [7671](#) PROPOSED STANDARD

Internet Engineering Task Force (IETF) Errata Exist  
Request for Comments: 6698 P. Hoffman  
Category: Standards Track VPN Consortium  
ISSN: 2070-1721 J. Schlyter  
Kirei AB  
August 2011

**The DNS-Based Authentication of Names  
Using Transport Layer Security**

Abstract

Encryption of the Internet often uses Transport Layer Security (TLS). This document depends on third parties to certify the keys used in that situation. This document improves on that situation by enabling the administrators of domain names to specify the keys used in that domain's TLS servers. This requires matching improvements in TLS client software, but no change in TLS server software.

Status of This Memo

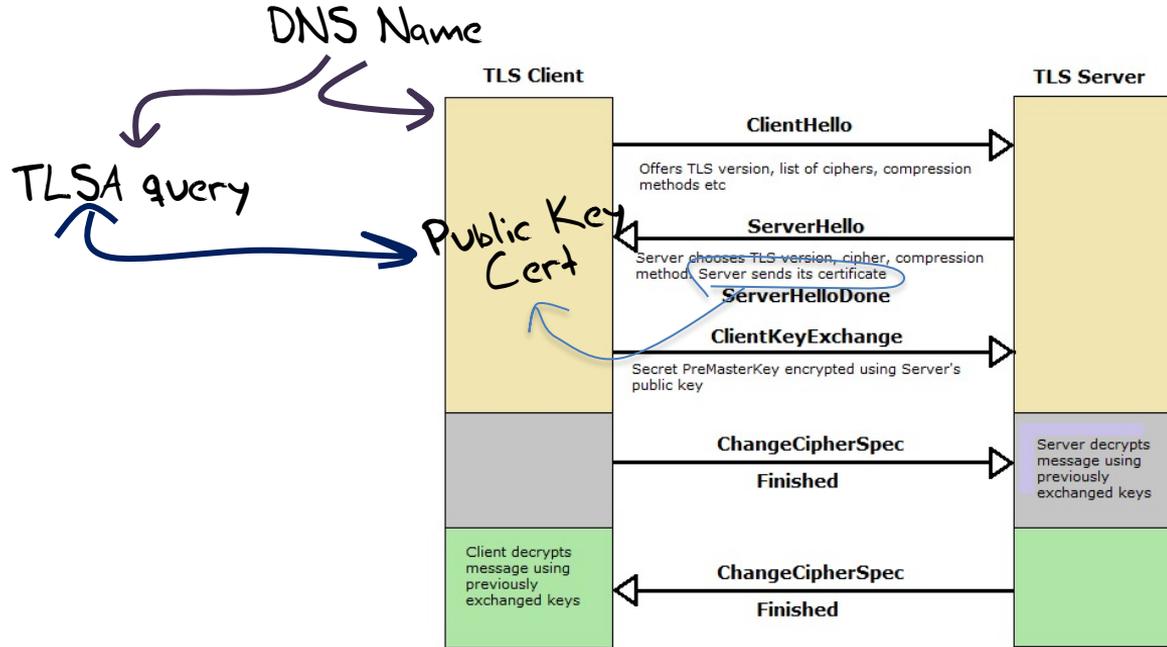
This is an Internet Standards Track document.

*RFC 6698 -- You should read this!*

# TLS with DANE

- Client receives server cert in Server Hello
  - *Client lookups the DNS for the TLSA Resource Record of the domain name*
  - *Client validates the presented certificate against the TLSA RR*
- Client performs Client Key exchange

# TLS Connections



# Just one problem...

- The DNS is full of liars and lies!
- And this can compromise the integrity of public key information embedded in the DNS
- Unless we fix the DNS we are no better off than before with these TLSA records!

# Just one response...

- We need to allow users to **validate** DNS responses for themselves
- And for this we need a Secure DNS framework
- Which we have – and it's called **DNSSEC!**

# DANE + DNSSEC

- Query the DNS for the TLSA record of the domain name and ask for the DNSSEC signature to be included in the response
- Validate the signature to ensure that you have an unbroken signature chain to the root trust point
- At this point you can accept the TLSA record as the authentic record, and set up a TLS session based on this data

# DANE + DNSSEC

- Query the DNS for the TLSA record of the domain name and ask for the DNSSEC signature to be included in the response
- Validate the signature to ensure that you have an unbroken signature chain to the root
- At this point you can *Yes, but No!* use the TLSA record as the authentic record, and set up a TLS session based on this data

# DANE + DNSSEC

## ImperialViolet

DNSSEC authenticated HTTPS in Chrome (16 Jun 2011)

**Update:** this has been removed from Chrome due to lack of use.

DNSSEC validation of HTTPS sites has been hanging around in Chrome for nearly a year now. But it's now enabled by default in the current canary and dev channels of Chrome and is on schedule to go stable with Chrome 14. If you're running a canary or dev channel (and you need today's dev channel release: 14.0.794.0) then you can go to <https://dnssec.imperialviolet.org> and see a DNSSEC signed site in action.



DNSSEC stapled certificates (and the reason that I use that phrase will become clear in a minute) are aimed at sites that currently have, or would use, self-signed certificates and, possibly, larger organisations that are Chrome based and want certificates for internal sites without having to bother with installing a custom root CA on all the client devices. Suggesting that this heralds the end of the CA system would be utterly inaccurate. Given the deployed base of software, all non-trivial sites will continue to use CA signed certificates for decades, at least. DNSSEC signing is just a gateway drug to better transport security.

DANE validation can be SO SLOW!

# Or...

## Faster validation?

[\[Docs\]](#) [\[txt|pdf\]](#) [\[draft-ietf-dnso...\]](#) [\[Tracker\]](#) [\[Diff1\]](#) [\[Diff2\]](#)

EXPERIMENTAL

Internet Engineering Task Force (IETF)  
Request for Comments: 7901  
Category: Experimental  
ISSN: 2070-1721

P. Wouters  
Red Hat  
June 2016

### **CHAIN Query Requests in DNS**

#### **Abstract**

This document defines an EDNS0 extension that can be used by a security-aware validating resolver configured to use a forwarding path along with the regular query answer. The reduction in queries potentially lowers the latency and reduces the need to send multiple queries at once. This extension mandates the use of source-IP-verified transport such as TCP or UDP with EDNS-COOKIE, so it cannot be abused in amplification attacks.

Status of This Memo

## Or ... Look! No DNS!

- Server packages server cert, TLSA record and the DNSSEC credential chain in a single bundle
- Client receives bundle in Server Hello
  - *Client performs validation of TLSA Resource Record using the supplied DNSSEC signatures plus the local DNS Root Trust Anchor without performing any DNS queries*
  - *Client validates the presented certificate against the TLSA RR*
- Client performs Client Key exchange

# Doing a better job

We could do a **far** better job at Internet Security:

- Publishing DNSSEC-signed zones

- Publishing DANE TLSA records

- Using DNSSEC-validating resolution

- Using TLSA records to guide TLS Key Exchange

- Stapling the TLSA + sig bundle into TLS

# Doing a better job

We could do a **far** better job

Publicly

*But nothing has happened for more than a decade!*

*Why not?*

→ Exchange

→ bundle into TLS

# Why is this so hard?

We have different goals?

- Some people want to provide strong hierarchical controls on the certificates and keys because it entrenches their role in providing services
- Some want to do it because it gives them a point of control to intrude into the conversation
- Others want to exploit weaknesses in the system to leverage a competitive advantage
- Some people think users prefer faster applications even if they have weaknesses
- Others think users are willing to pay a time penalty for better authentication controls

# Why is this so hard?

Because there are so many moving parts?

- In a system that is constructed upon the efforts of multiple systems and multiple providers we are relying on some one in charge to orchestrate the components to as working whole



Saturn V Launch Vehicle

Three stage rocket, each built by a different contractor

Each of whom used multiple subcontractors

3 million components

Each supplied by the lowest bidder!

# Why is this so hard?

Because we are relying on the market to provide coherence and consistency of orchestration across providers?

- And perhaps that's the key point here
- Loosely coupled systems will always present windows of vulnerability
  - Routing integrity
  - Name registration
  - Name certification
  - Service control
- Effective defence involves not only component defence but also in defending the points of interaction between components
- And we find this very hard to achieve when the market itself is the orchestration agent

# Users and Trust

- Users just want to be able to trust that the websites and services that they connect to and share their credentials, passwords and content with are truly the ones they expected to be using without first studying for a PhD in Network Operational Security
- Somehow we're missing that simple objective and we've interposed complexity and adornment that have taken on a life of their own and are in fact eroding trust
- And that's bad!
- If we can't trust our communications infrastructure, then we don't have a useful communications infrastructure.

What a dysfunctional mess we've created!

That's it!

Questions?