# Security

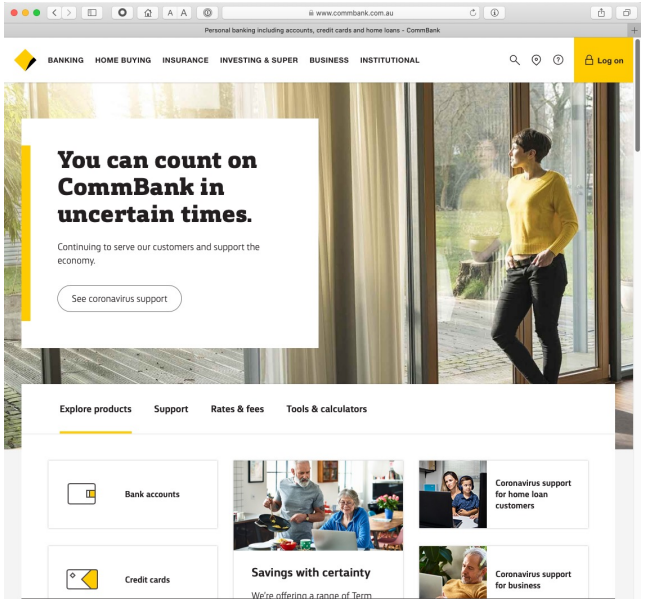
Geoff Huston
Chief Scientist, APNIC

~~Security~~

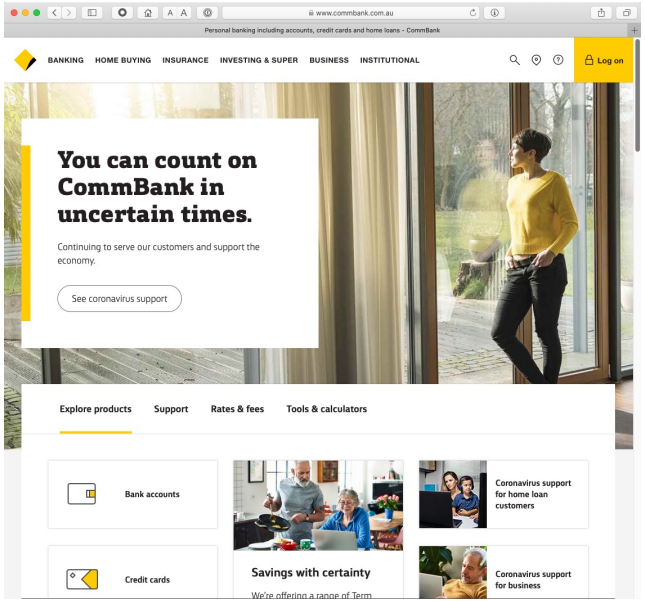
'insecurity!

Geoff Huston
Chief Scientist, APNIC

# Which Bank?

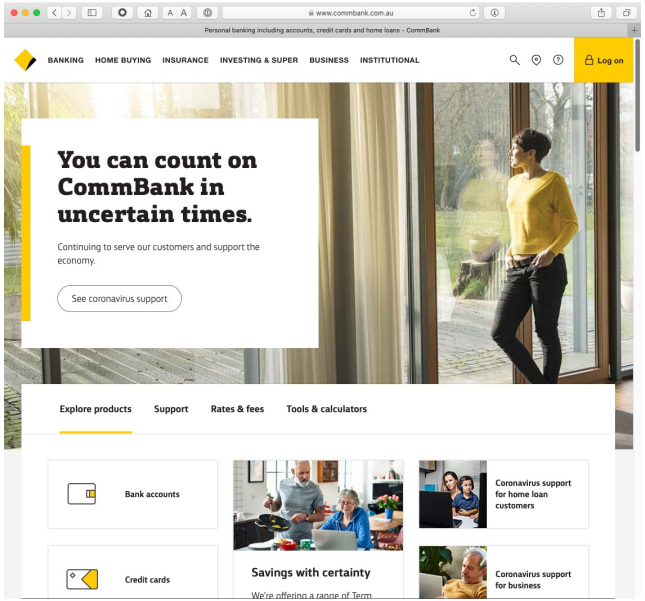

www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank
BANKING
HOME BUYING
INSURANCE
INVESTING & SUPER
BUSINESS
INSTITUTIONAL
Log on
You can count on CommBank in uncertain times.
Continuing to serve our customers and support the economy.
See coronavirus support
Explore products
Support
Rates & fees
Tools & calculators
Bank accounts
Credit cards
Savings with certainty
Coronavirus support for home loan customers
Coronavirus support for business

# Which Bank? My Bank!



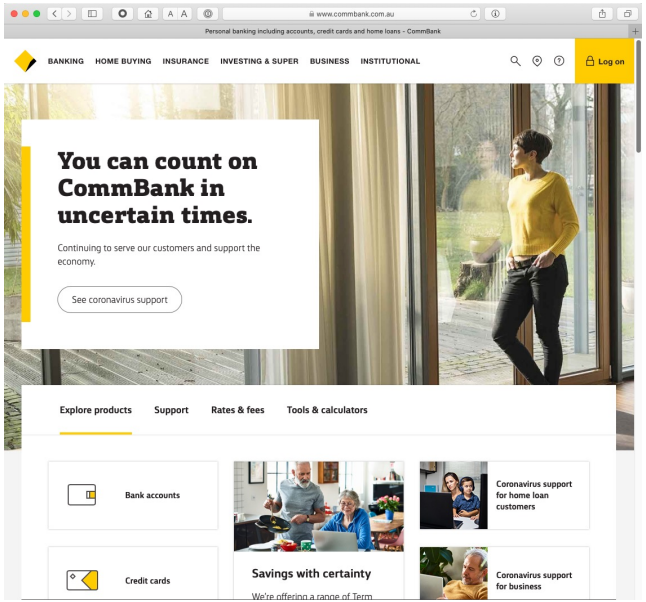
# Which Bank? My Bank!

i hope!



# Security on the Internet

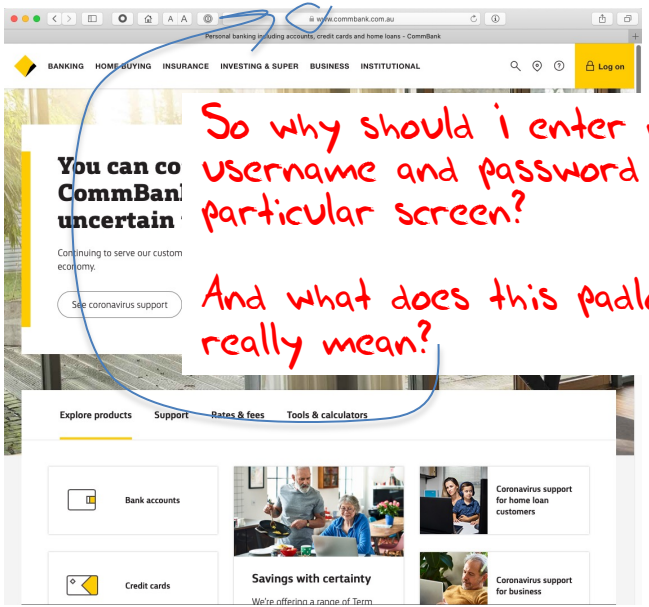
How do you know that you are really going to where you thought you were going to?




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?




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

And what does this padlock icon really mean?

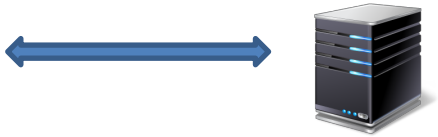
Its trivial to mock up a web page to look like another

# 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…

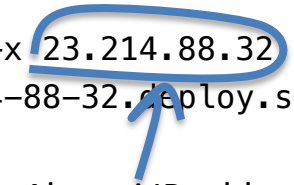
```
$ dig -x 23.214.88.32 +short
a23-214-88-32.deploy.static.akamaitechnologies.com.
```

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!

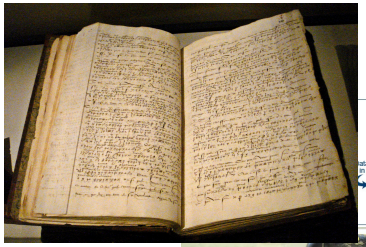
Why should my browser trust that 23.214.88.32 is really the authentic web site for the Commonwealth Bank of Australia, and not some dastardly evil scam designed to steal my passwords and my money?

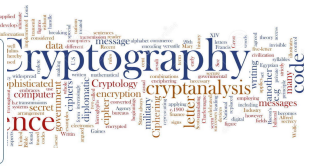
And why should I trust my browser?

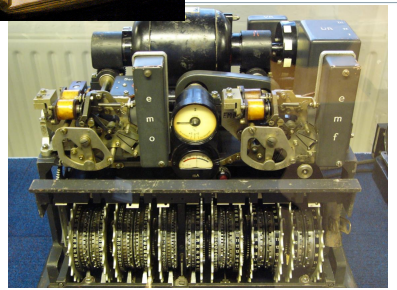
# The major question…

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

# It's all about cryptography



# 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

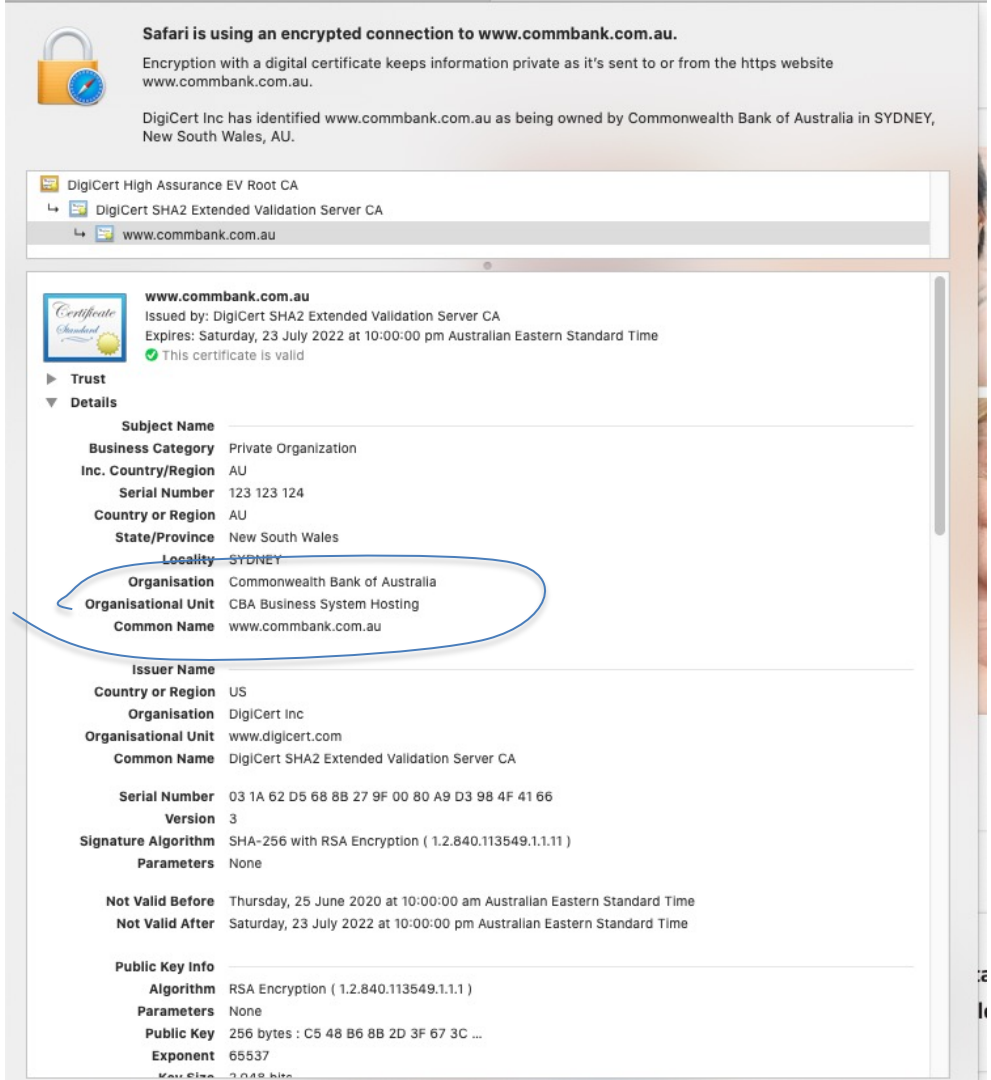
What if I ‘trust’ an intermediary?

- 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.**

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

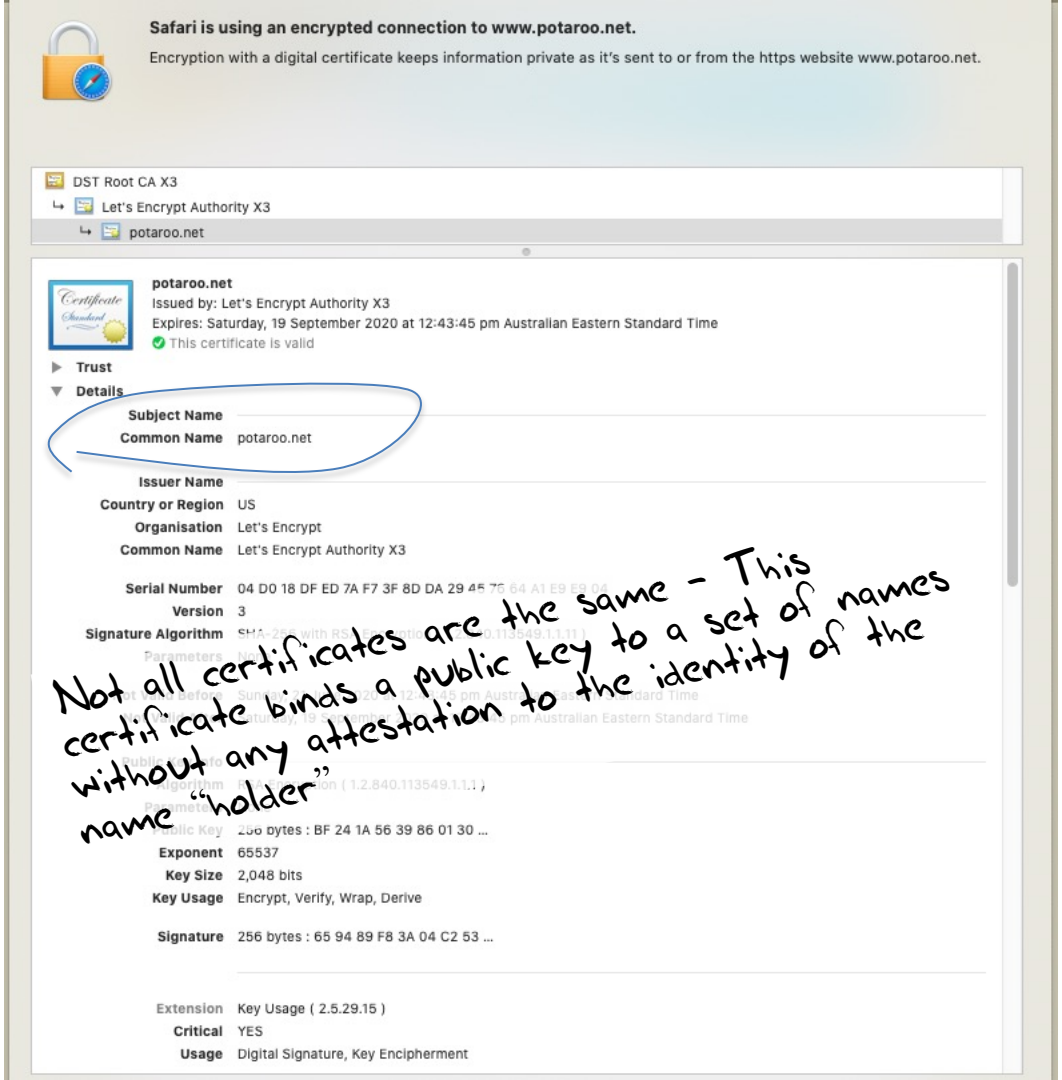
DigiCert Inc has identified www.commbank.com.au as being owned by Commonwealth Bank of Australia in SYDNEY, New South Wales, AU.

DigiCert High Assurance EV Root CA
↳ DigiCert SHA2 Extended Validation Server CA
↳ www.commbank.com.au

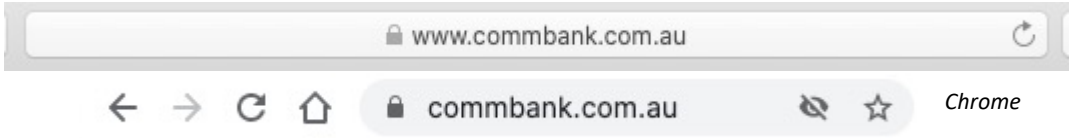
**www.commbank.com.au**
Issued by: DigiCert SHA2 Extended Validation Server CA
Expires: Saturday, 23 July 2022 at 10:00:00 pm Australian Eastern Standard Time
This certificate is valid

▶ **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 ( 1.2.840.113549.1.1.11 ) |
| **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 |

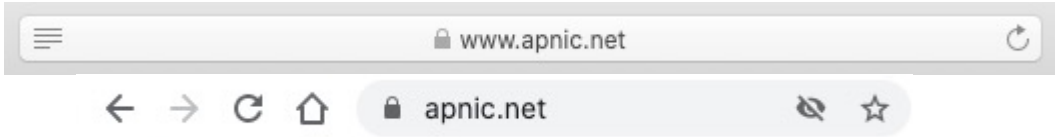

Safari is using an encrypted connection to www.potaroo.net.
Encryption with a digital certificate keeps information private as it's sent to or from the https website www.potaroo.net.
DST Root CA X3
Let's Encrypt Authority X3
potaroo.net
potaroo.net
Issued by: Let's Encrypt Authority X3
Expires: Saturday, 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
Version 3
Signature Algorithm
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"

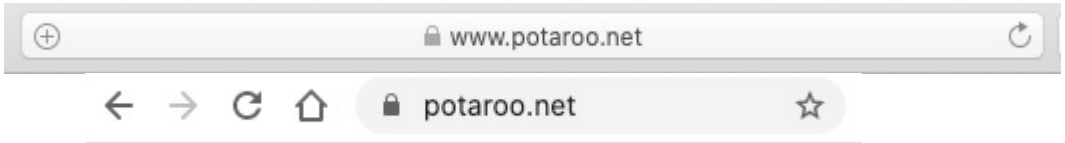
# Spot the Difference



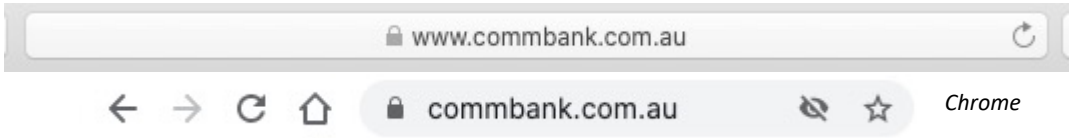

*Safari*

*Chrome*



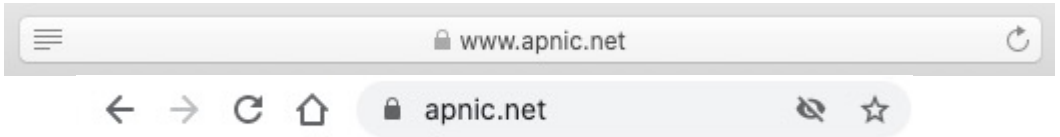
# Spot the Difference



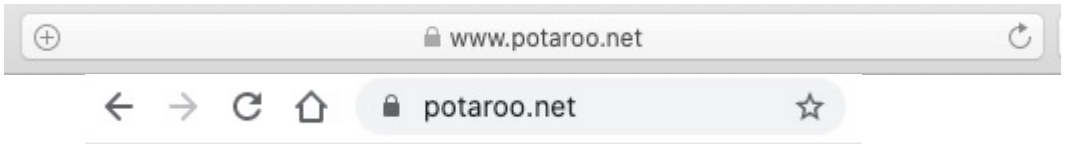

*Safari*

*Chrome*

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!!

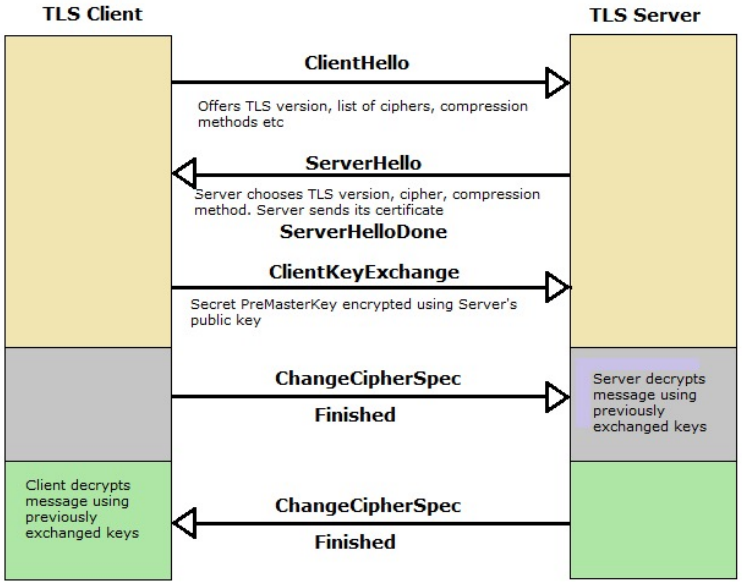



This web site's certificate says *nothing* about the entity that holds the public key associated with this domain

# Moving on…

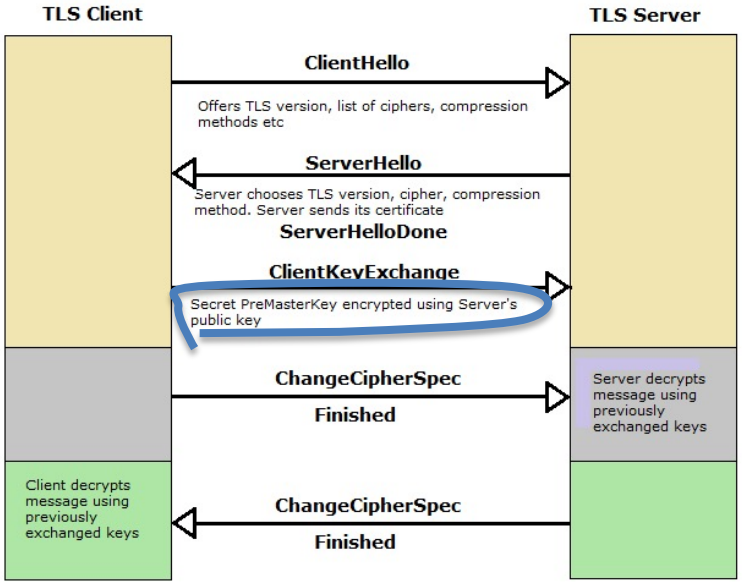
- Ok, so the certificate system is a total mess, but TLS still works, right?

# Secure Connections using TLS



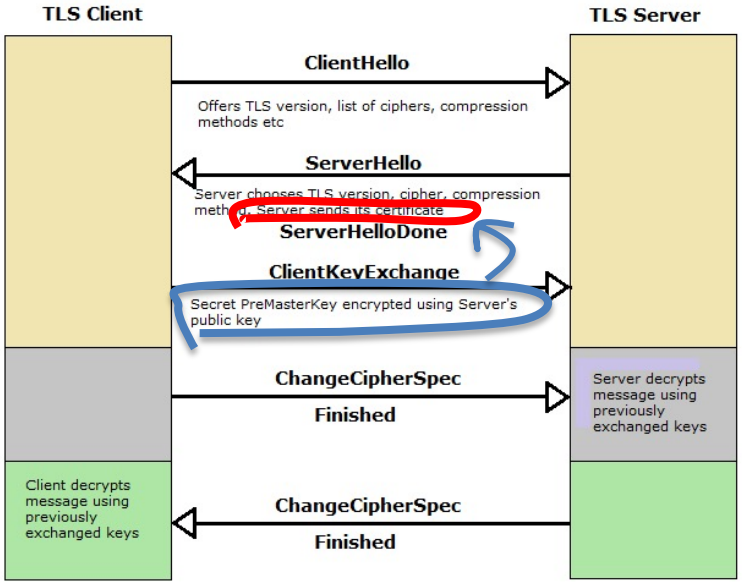

https://rhsecurity.wordpress.com/tag/tls/

# Secure Connections using TLS

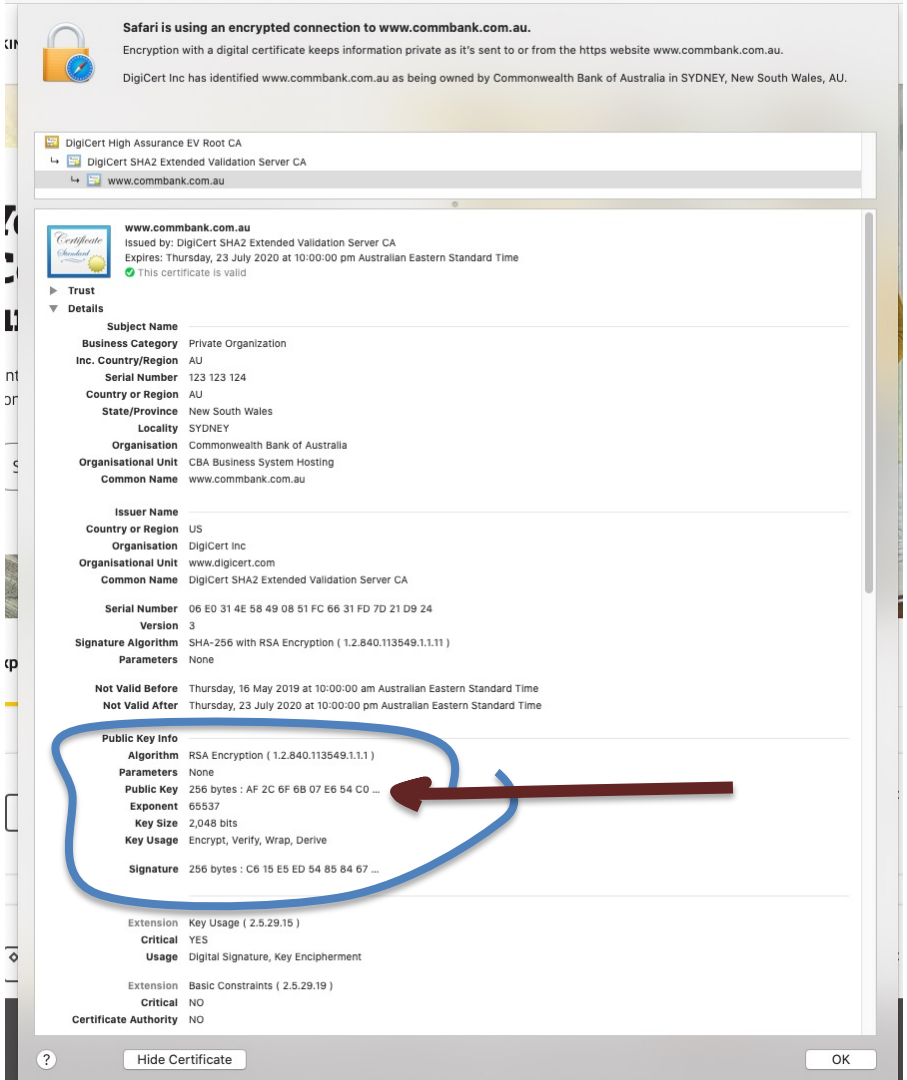



https://rhsecurity.wordpress.com/tag/tls/

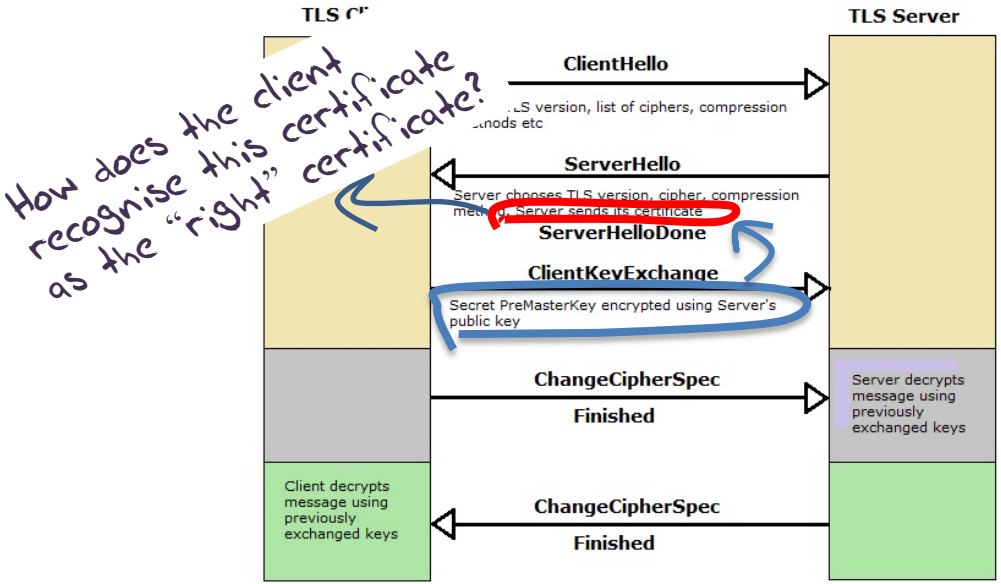
# Secure Connections using TLS



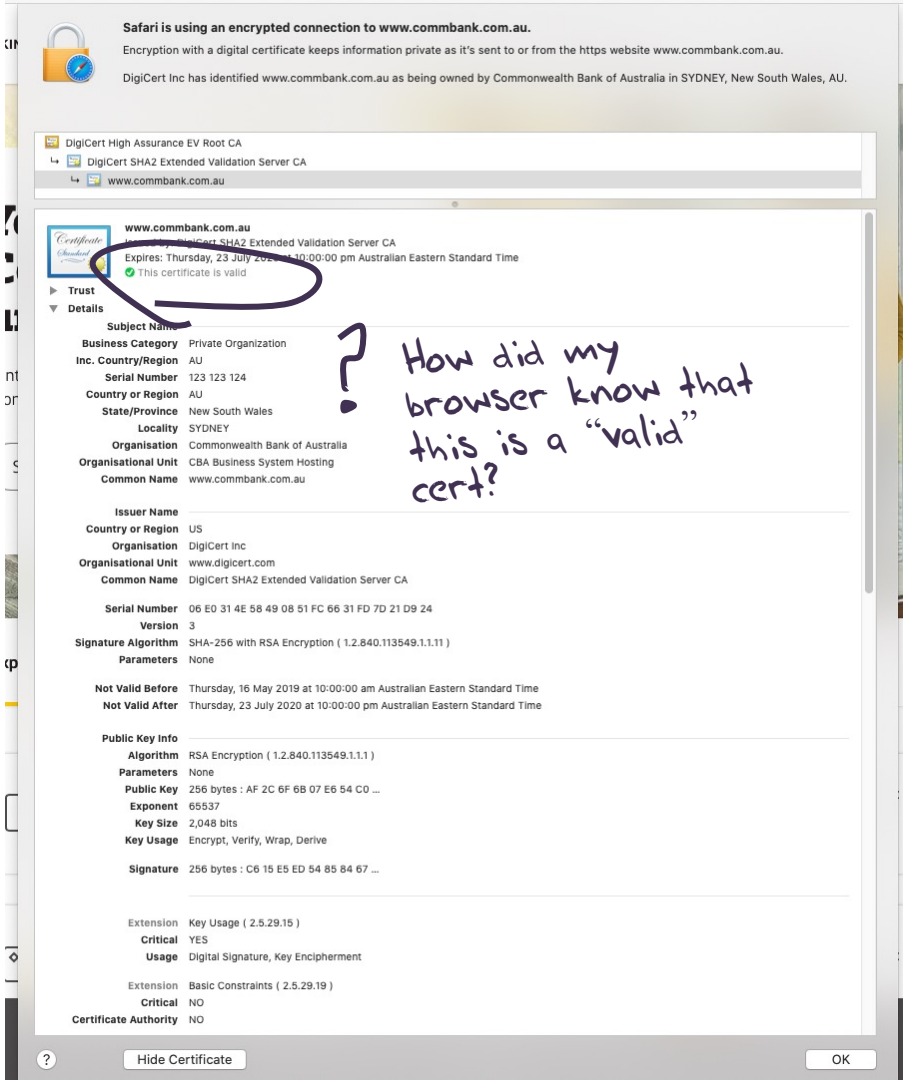

https://rhsecurity.wordpress.com/tag/tls/


Safari is using an encrypted connection to 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.
DigiCert Inc has identified 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
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
Issuer Name
Country or Region US
Organisation DigiCert Inc
Organisational Unit 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 ( 1.2.840.113549.1.1.11 )
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
Hide Certificate
OK

# Secure Connections using TLS




https://rhsecurity.wordpress.com/tag/tls/


Safari is using an encrypted connection to 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.
DigiCert Inc has identified 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
www.commbank.com.au
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
Issuer Name
Country or Region US
Organisation DigiCert Inc
Organisational Unit 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 ( 1.2.840.113549.1.1.11 )
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 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, 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!

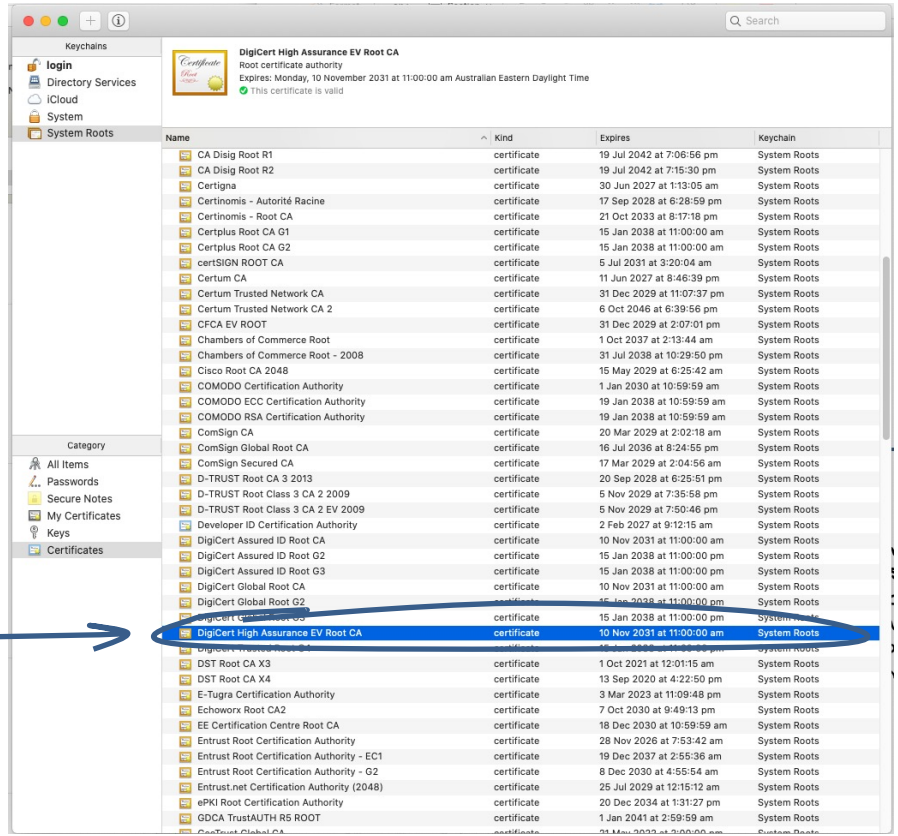
# Domain Name Certification

- 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 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, as long as I am prepared to trust DigiCert and the certificates that they issue
- And I’m , them because DigiCert NEVER lie!

How do i know that? Why should i trust them?

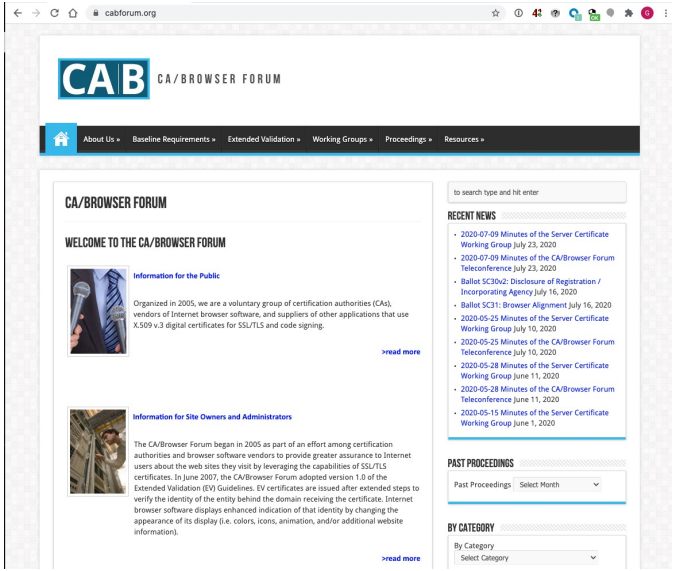
# Local Trust

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



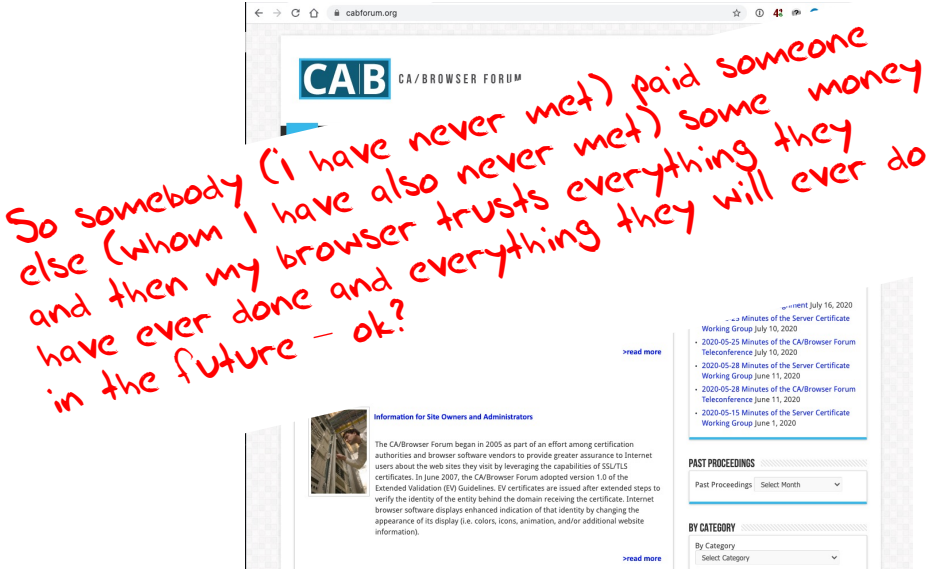
# Local Trust

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!



# Local Trust

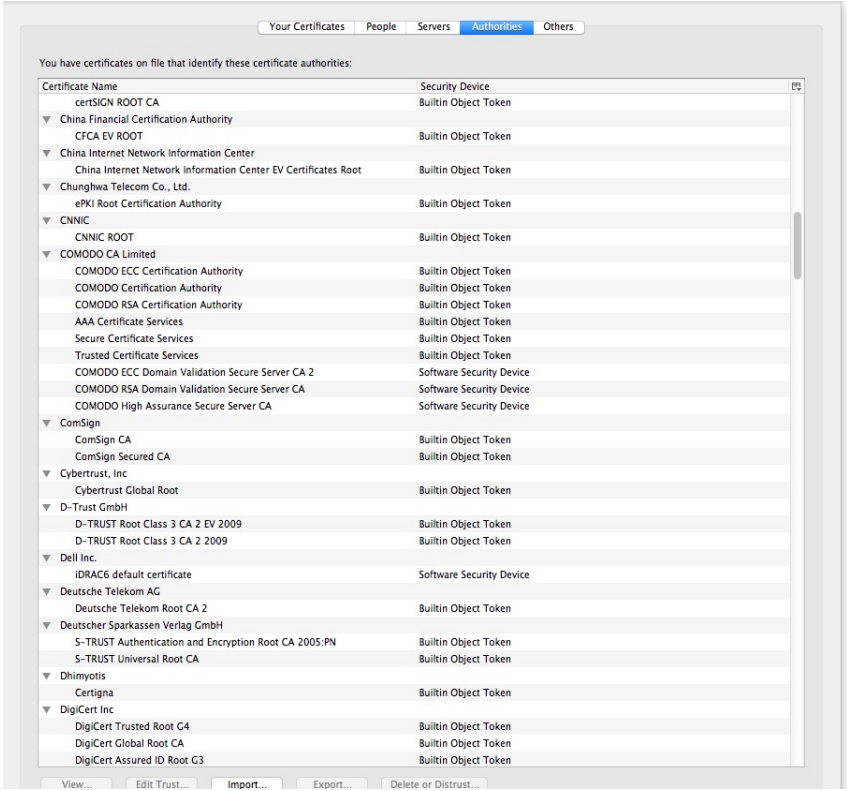
These Certificate Authorities are listed in my computer's trust set because they claim to operate according to the practices defined by the CAB industry forum (of which they are a member) and they **never** lie!



# Local Trust or Local Credulity*?

Wow!

Are they **all** trustable?



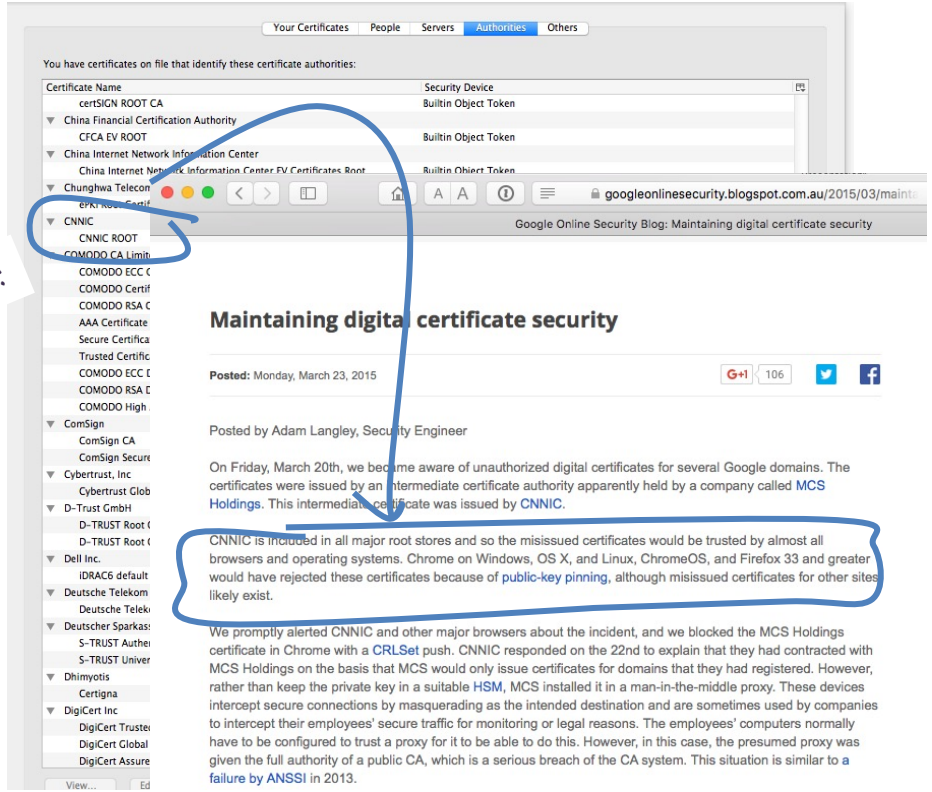

* cre·du·li·ty
/krəˈd(y)o͞oləd̄ē/
*noun*
a tendency to be too ready to believe that something is real or true.

# Local Credulity

Wow!

Are they **all** trustable?

Evidently Not!

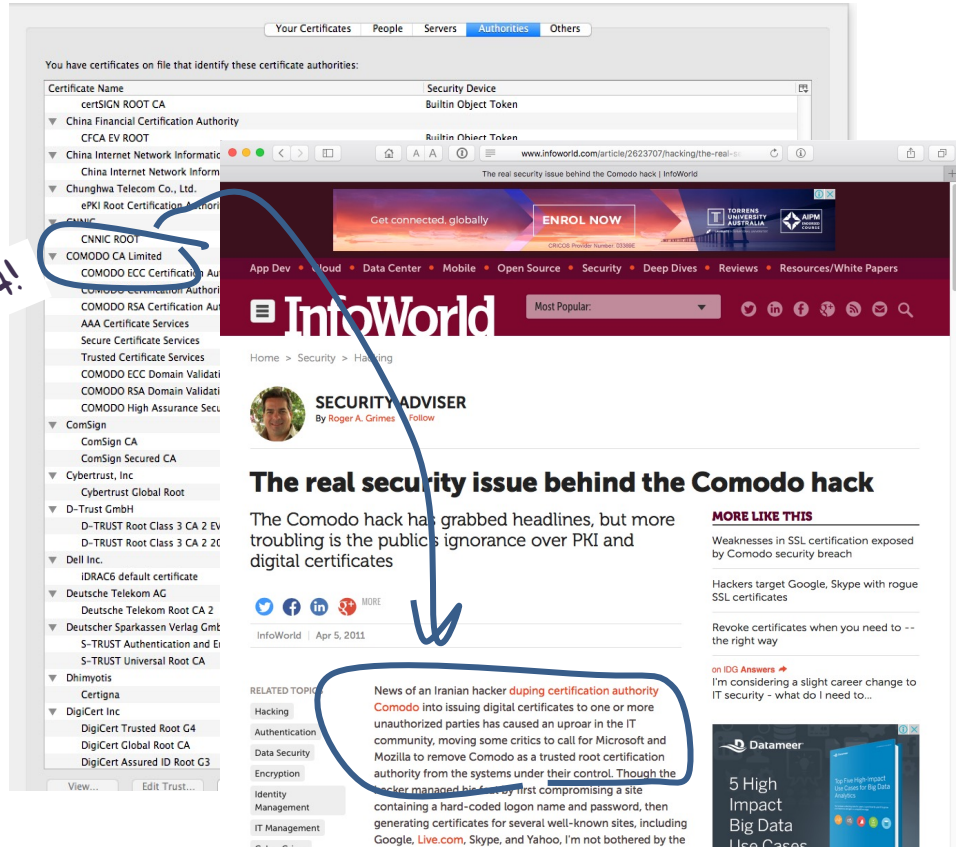


# Local Credulity

Wow!

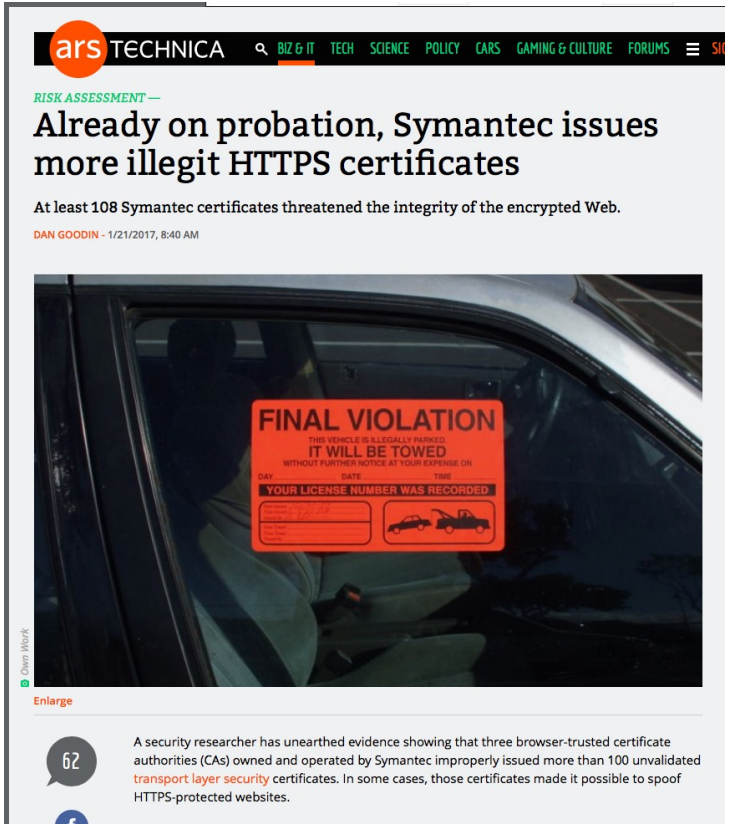
Are they **all** trustable?

Evidently Not!



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



Own Work

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=A8F14F52CC12B2D7153A13316E7DA39E6AE37B1A10C16288B9024A9B9DC3C4C6

https://crt.sh/?sha256=8B5956C57FDCF720B6907A4B1BC8CA2E46CD90EAD5C061A426CF48A6117BFBFA

https://crt.sh/?sha256=94482136A1400BC3A1136FECA3E79D4D200E03DD20B245D19F0E78B5679EAF48

https://crt.sh/?sha256=C69AB04C1B20E6FC7861C67476CADDA1DAE7A8DCF6E23E15311C2D2794BFCD11

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

# With unpleasant consequences when it all goes wrong

sters helped ignited
untry's 45-member

talk
ow

Cuba aimed at U.S.
er husband not to
anything happens,
tay right here with
told him in October
be with you, and I
u, and the children
ithout you."
nterview conducted
e of only three that
after Mr. Kennedy's
ublished as a

ng television interview.

Société Générale, BNP Paribas and Crédit Agricole, are considered integral actors in the French economy, lending

VOLATILITY IS THE NEW MARKET NORM
Large swings in share prices are more common now than at any other time in recent stock market history. PAGE 16

## Iranian activists feel the chill as hacker taps into e-mails

BY SOMINI SENGUPTA

He claims to be 21 years old, a student of software engineering in Tehran who reveres Ayatollah Ali Khamenei and despises dissidents in his country.

He sneaked into the computer systems of a security firm on the outskirts of Amsterdam. He created fake credentials that could allow someone to spy on Internet connections that appeared to be secure. He then shared that bounty with people he declines to identify.

The fruits of his labor are believed to
ed to tap into the online
many as 300,000
ummer.

online security mechanism that is trusted by Internet users all over the world.

Comodohacker, as he calls himself, insists that he acted on his own and is unperturbed by the notion that his work might have been used to spy on anti-government compatriots.

"I'm totally independent," he said in an e-mail exchange with The New York Times. "I just share my findings with some people in Iran. They are free to do anything they want with my findings and things I share with them, but I'm
ible."
ternet attacks, this
oment of
In the
is most
reckon
HACKER, PA

International Herald Tribune
Sep 13, 2011 Front Page

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


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 sh
di
pu
- Th
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!

# 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?

# 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.

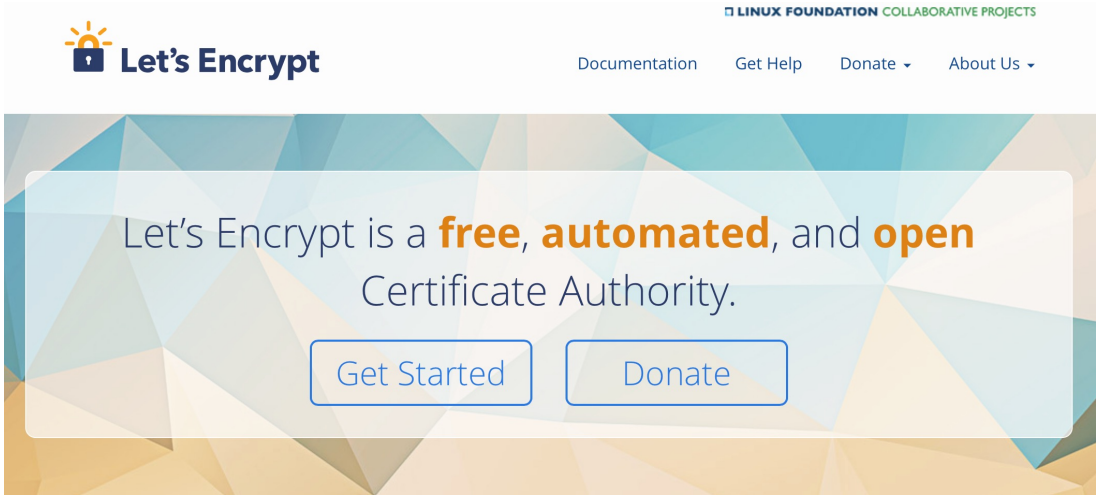
| Platform | Chrome | Firefox | Opera | Safari | Edge |
|---|---|---|---|---|---|
| **Mac OS X** 10.15.3 | **YES** 80.0.3987.132 | **YES** 73.0.1 | **YES** 67.0.3575.53 | **YES** 13.0.5 | |
| **iOS** 13.3.1 | **YES** 80.0.3987.95 | **YES** 23.0 | **NO** 16.0.15 | **YES** 13.3.1 | |
| **Android** 10 | **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*

https://www.potaroo.net/ispcol/2020-03/revocation.html

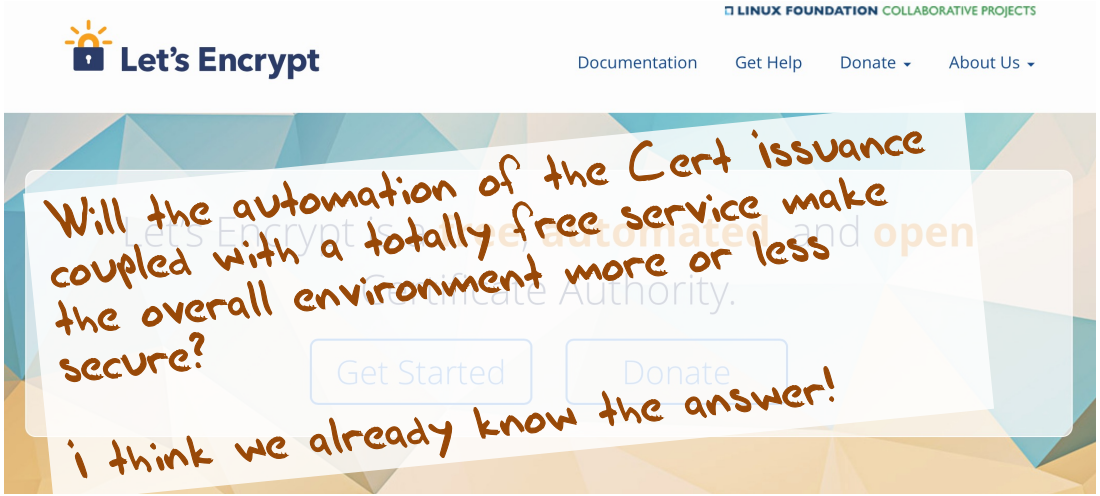
# How can we fix this?

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



# How can we fix this?

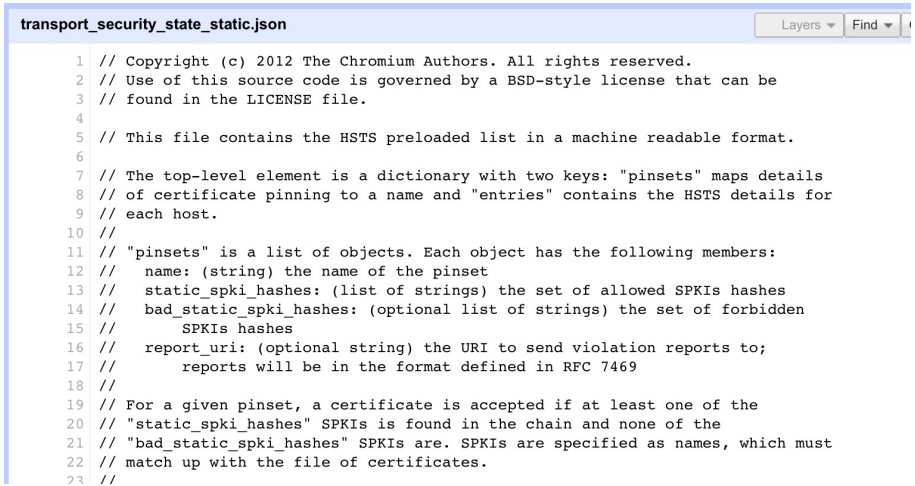
Option A:  Take all the money out of the system!



# 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



# 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

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

its just Google protecting itself and no one else

transport_security_state_static.json

```
// found in the LICENSE file.

// This file contains the HSTS preloaded list in a machine readable format.

// The top-level element is a dictionary with two keys: "pinsets" maps details
// of certificate pinning to a name and "entries" contains the HSTS details for
// each host.
//
// "pinsets" is a list of objects. Each object has the following members:
//   name: (string) the name of the pinset
//   static_spki_hashes: (list of strings) the set of allowed SPKIs hashes
//   bad_static_spki_hashes: (optional list of strings) the set of forbidden
//       SPKIs hashes
//   report_uri: (optional string) the URI to send violation reports to;
//       reports will be in the format defined in RFC 7469
//
// For a given pinset, a certificate is accepted if at least one of the
// "static_spki_hashes" SPKIs is found in the chain and none of the
// "bad_static_spki_hashes" SPKIs are. SPKIs are specified as names, which must
// match up with the file of certificates.
//
```

# How can we fix this?

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

its just Google protecting itself and no one else

Option B: White Listing and Pinning with HSTS

https://code.google.com/p/chromium/codesearch#cnromium/src/net/http/transport_security_state_static.json

INFOWORLD TECH WATCH

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

About | Informed news analysis every weekday

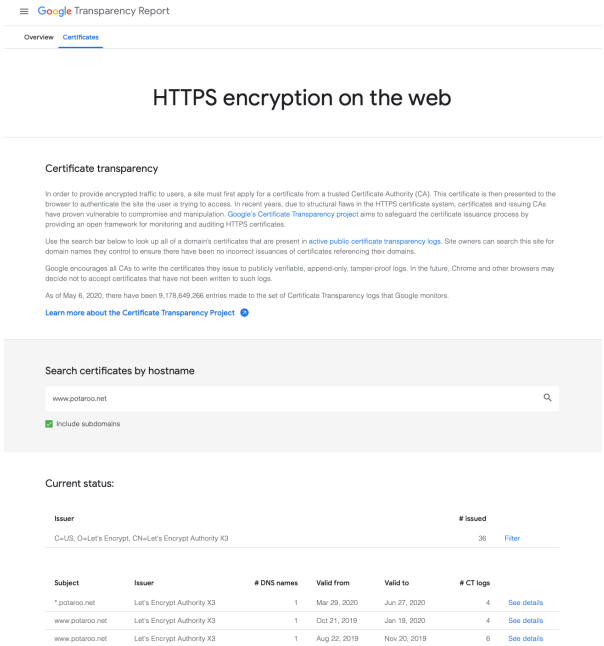
## Google moves into the Certificate Authority business

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 //
```

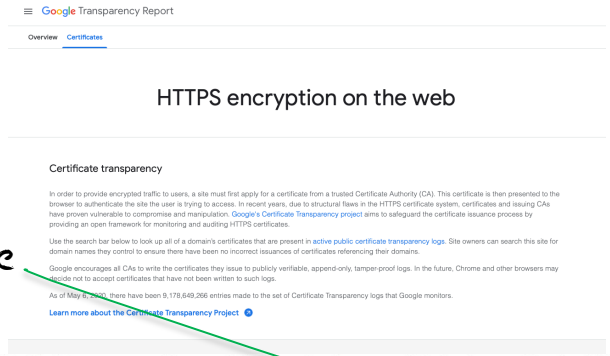
# How can we fix this?

## Option C: Certificate Transparency



# How can we fix this?

## Option C: Certificate Transparency




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:

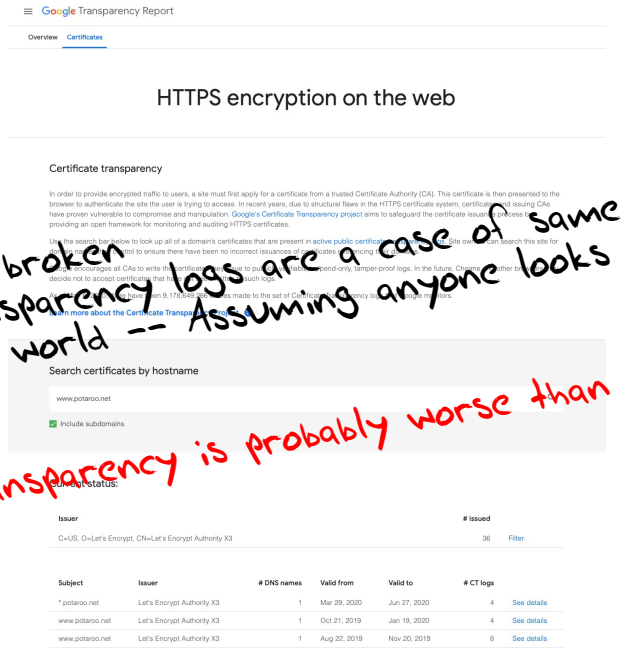
| Issuer | # Issued | |
|---|---|---|
| C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3 | 36 | Filter |

| 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 | See details |
| www.potaroo.net | Let's Encrypt Authority X3 | 1 | Oct 21, 2019 | Jan 19, 2020 | 4 | See details |
| www.potaroo.net | Let's Encrypt Authority X3 | 1 | Aug 22, 2019 | Nov 20, 2019 | 6 | See details |

This is a fail

# How can we fix this?

## Option C: Certificate Transparency

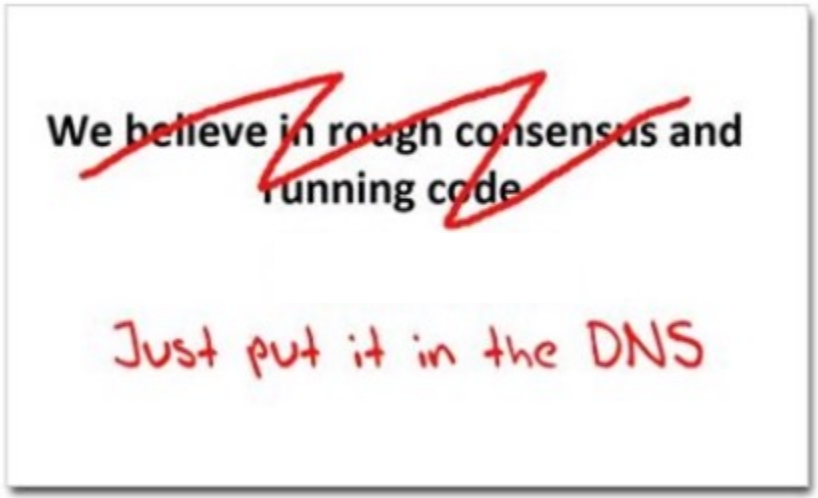



its just so broken
These transparency logs are a case of same week service in a millisecond world -- Assuming anyone looks in the first place!

Cert Transparency is probably worse than a placebo!

# 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?

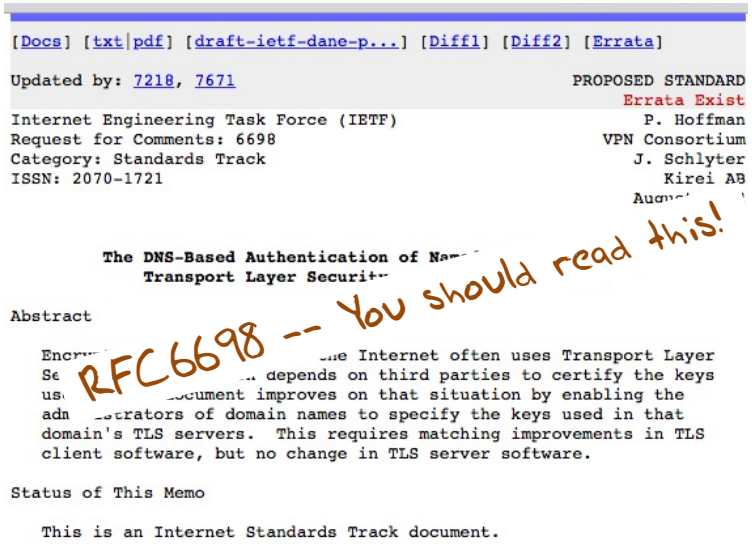
Where better to find out the public key associated with a DNS-named service than to look it up in the DNS?

- Why not query the DNS for the HSTS…
- Why not query the DNS for…er CA?
- Why not query the…ne hash of the domain name cert?
- Why not…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, Errata Exist

Internet Engineering Task Force (IETF)
Request for Comments: 6698
Category: Standards Track
ISSN: 2070-1721

P. Hoffman, VPN Consortium
J. Schlyter, Kirei AB
Aug...

**The DNS-Based Authentication of Na...**
**Transport Layer Securi...**

RFC6698 -- You should read this!

Abstract

Encr... ...e Internet often uses Transport Layer Se... ... depends on third parties to certify the keys us... ...ocument improves on that situation by enabling the adm... ...erators 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.

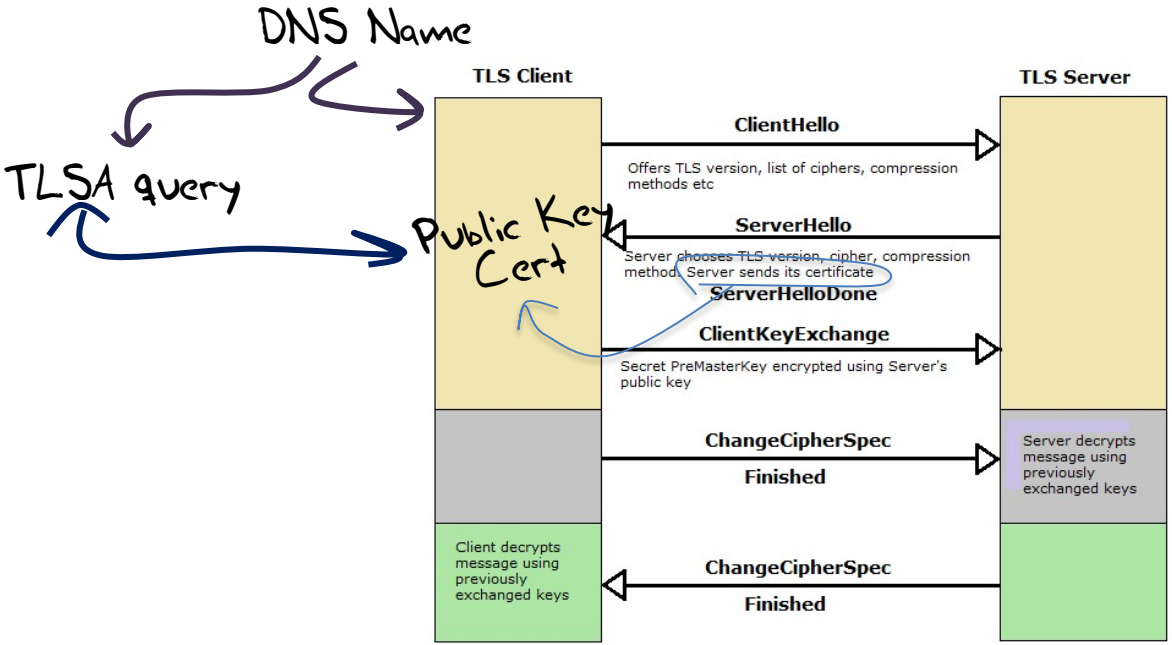
Status of This Memo

This is an Internet Standards Track document.

# 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




https://rhsecurity.wordpress.com/tag/tls/

# 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 use the TLSA record as the authentic record, and set up a TLS session based on this data

Yes, but No!

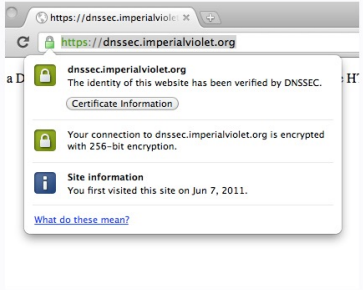
# DANE + DNSSEC

## 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.

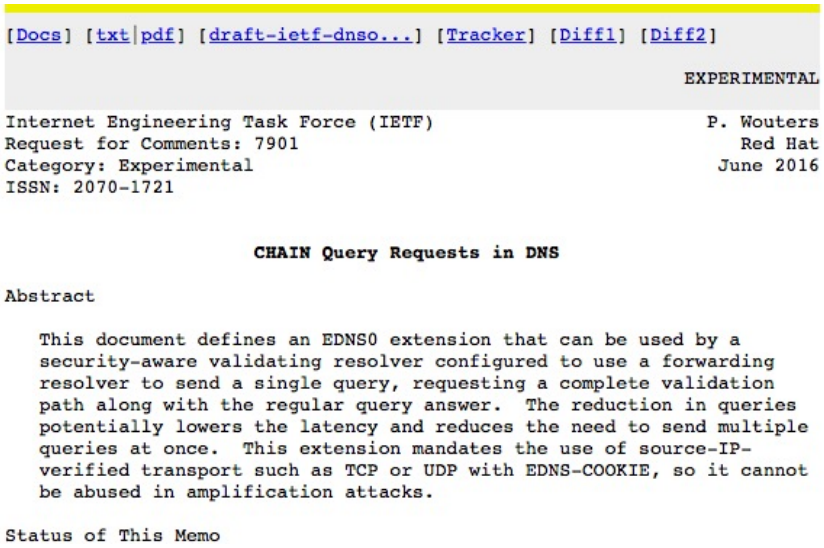



DNSSEC stapled certificates (and the reason that I use that phrase will become clear in a minute) are aimed at sites that currently have, or would use, self-signed certificates and, possibly, larger organisations that are Chrome based and want certificates for internal sites without having to bother 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-trival 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) — P. Wouters
Request for Comments: 7901 — Red Hat
Category: Experimental — June 2016
ISSN: 2070-1721

**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 resolver to send a single query, requesting a complete validation 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 DNSEC 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** bette

Publi

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?