The Foundations of Network Security

Geoff Huston AM
Chief Scientist, APNIC

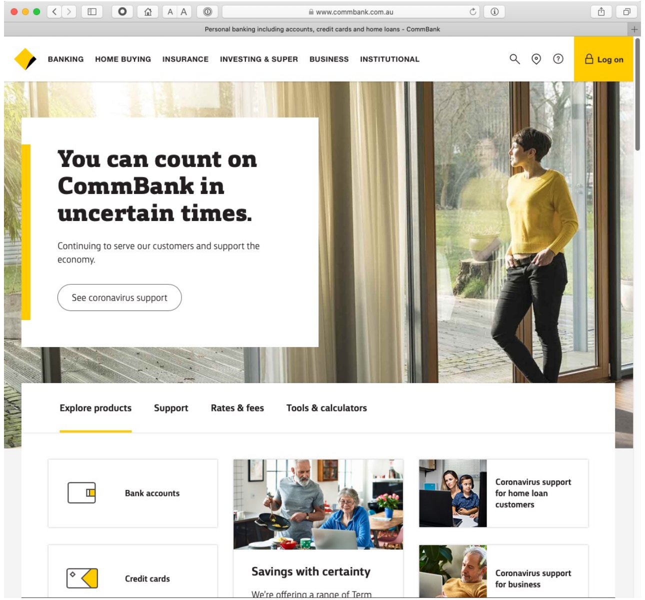
The Foundations of Network ~~Security~~

insecurity!

Geoff Huston AM
Chief Scientist, APNIC

1. The Current Security Framework for the Internet

Which Bank?

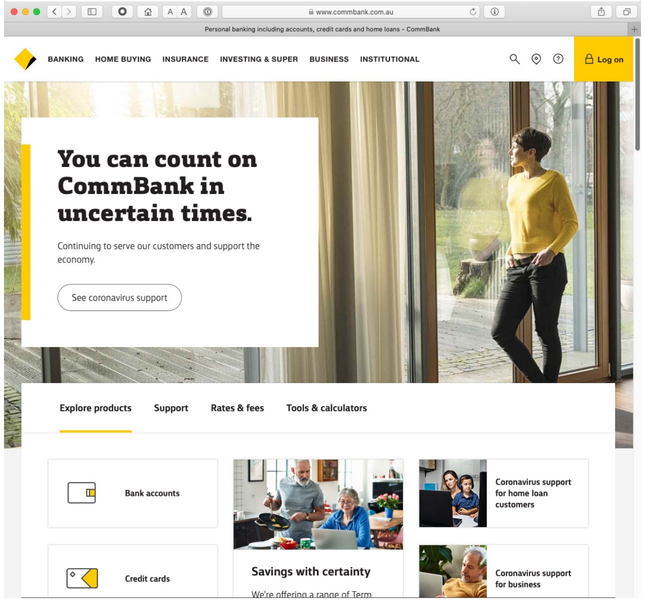


Let’s start with a simple example:

Why should you pass your account and password to this web site? It might look like your bank, but frankly it could just as easily be a fraudulent site intended to steal your banking credentials. Why should you trust what you see on the screen?

Which Bank? My Bank!

i hope!

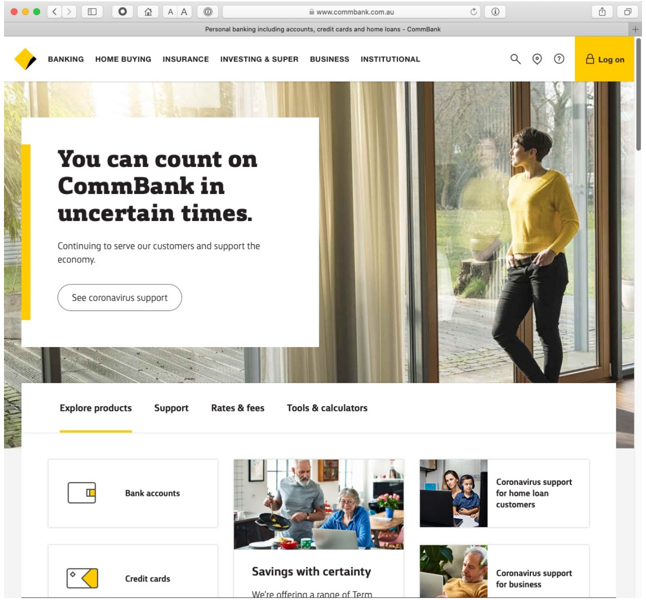


Ok – its not a random example. It’s the online bank I use! But the same question is still there. Why should I trust this web page?

Security on the Internet

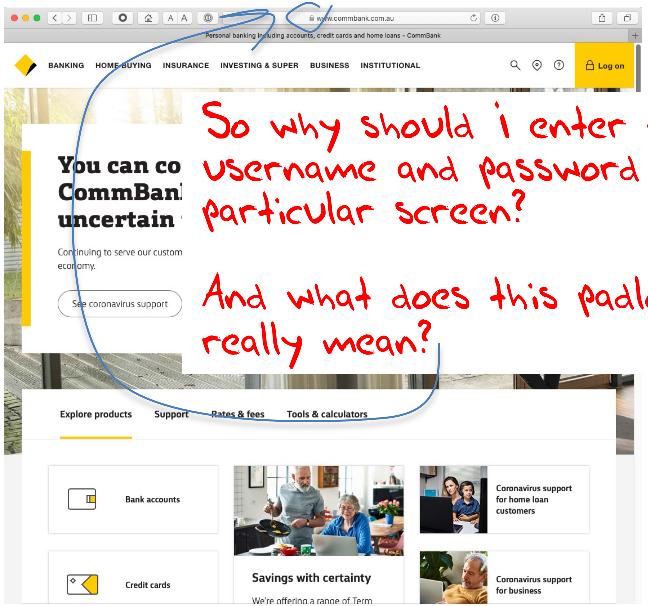
How do you know that you are really going to where you thought you were going to?

its trivial to create a web page to look exactly 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 create a web page to look exactly like another

Opening the Connection: First Steps

Client:

DNS Query:

www.commbank.com.au?

DNS Response:

104.116.164.218

TCP Session:

TCP Connect 104.116.164.218, port 443

Hang on…

Who “owns” that IP address? The Commonwealth Bank? Someone else?

Let’s look at little more:

```
$ dig -x 104.116.164.218 +short
a104-116-164-218.deploy.static.akamaitechnologies.com
```

Hang on…

```
$ dig -x 104.116.164.218 +short
a104-116-164-218.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 104.116.164.218 +short @::1
a104-116-164-218.deploy.static.akamaitechnologies.com
```

That’s an Akamai IP address

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

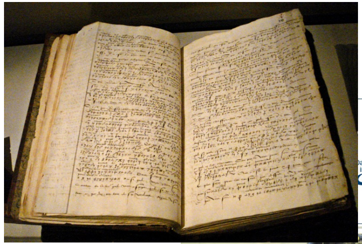
Why should my browser trust that 104.116.164.218 is really the authentic web site for the Commonwealth Bank of Australia, and not some dastardly evil scam designed to steal my passwords and my money?

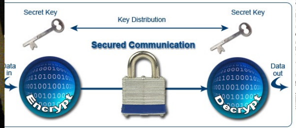
And why should I trust my browser?

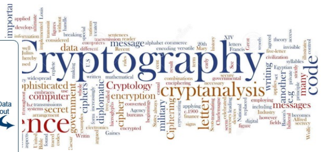
The major question…

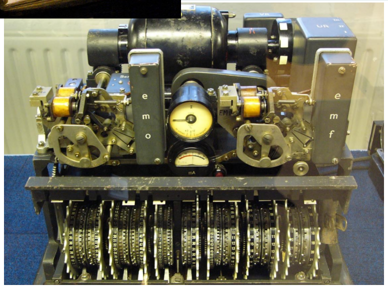
How does my browser tell the difference between an intended truth and a dastardly lie?

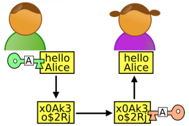
It's all about cryptography











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

This is important

So I will repeat it:

- Using public/private key cryptography requires a pair of keys (A,B) such that:
  - Anything encrypted using key A can ONLY be decrypted using key B, and no other key
  - Anything encrypted using key B can ONLY be decrypted using key A, and no other key
  - Knowing the value of one key WILL NOT let you work out the value of the other key anytime soon!

This form of asymmetric cryptography lies at the heart of the Internet's security framework

Public/Private Key Pairs

If I have a copy of your PUBLIC key, and you encrypt a message with your PRIVATE key, and I can decrypt the message using your PUBLIC key, then

- I know no one has tampered with your original message
- And I know it was you that sent it.
- And you can't deny it.

If we negotiate a session key using the combination of your public key and a local private session key and encrypt all session messages using this session key, then

- I am confident no one else can eavesdrop on our conversation in this session

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!

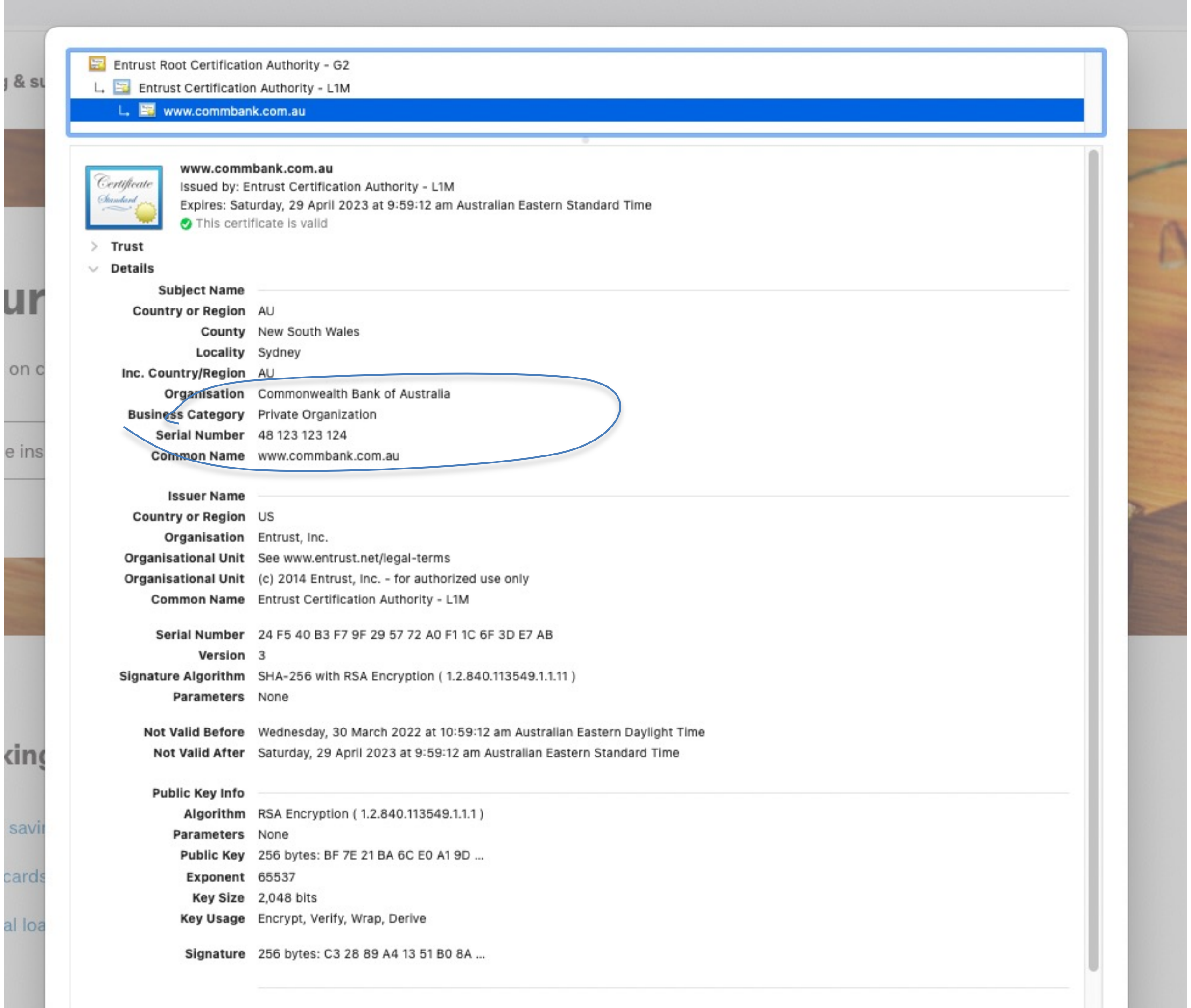
Public Key Certificates

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

- If this trusted 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"

** If you have ever used "public notaries" to validate a document, then this is a digital equivalent*

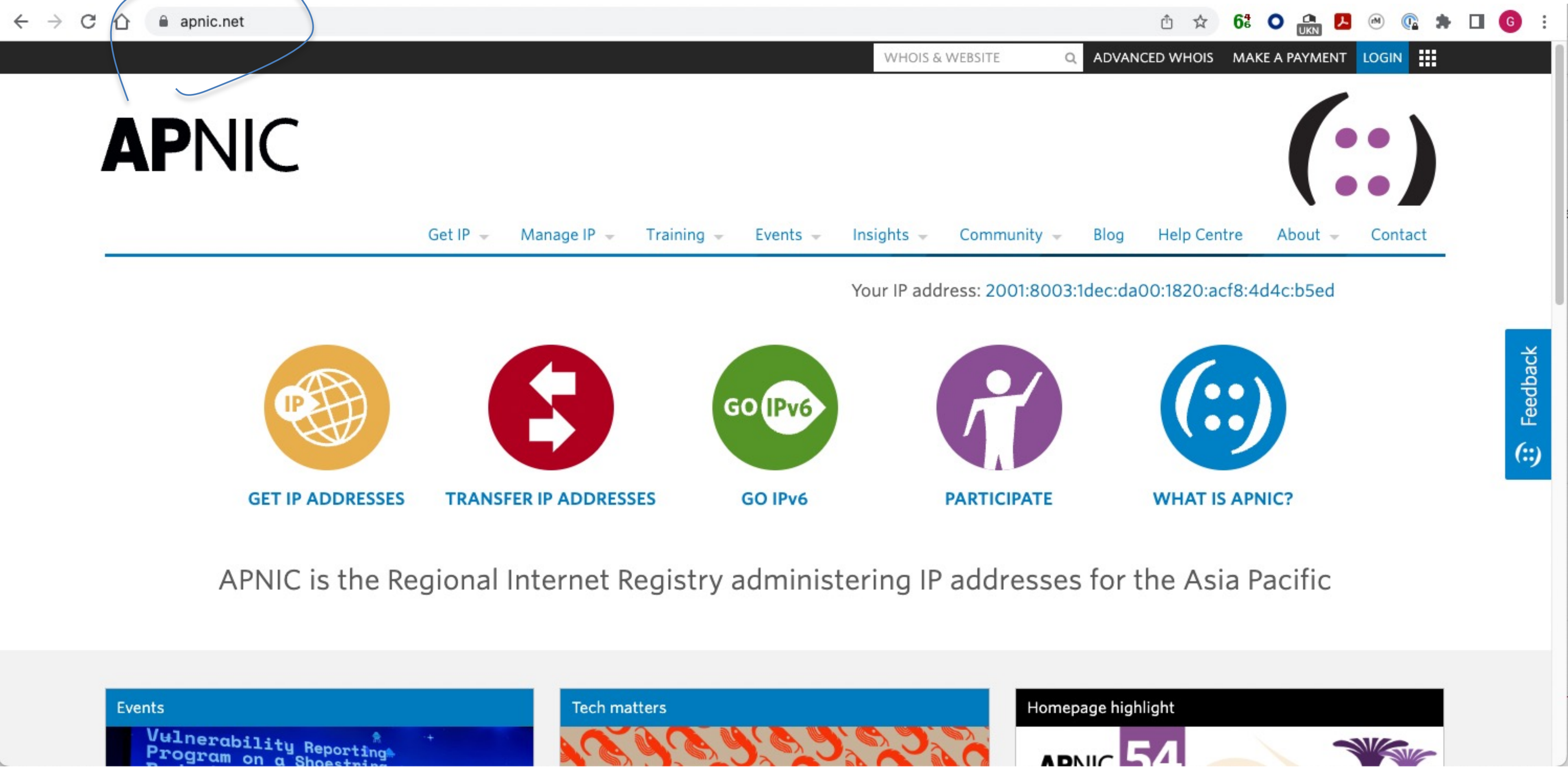


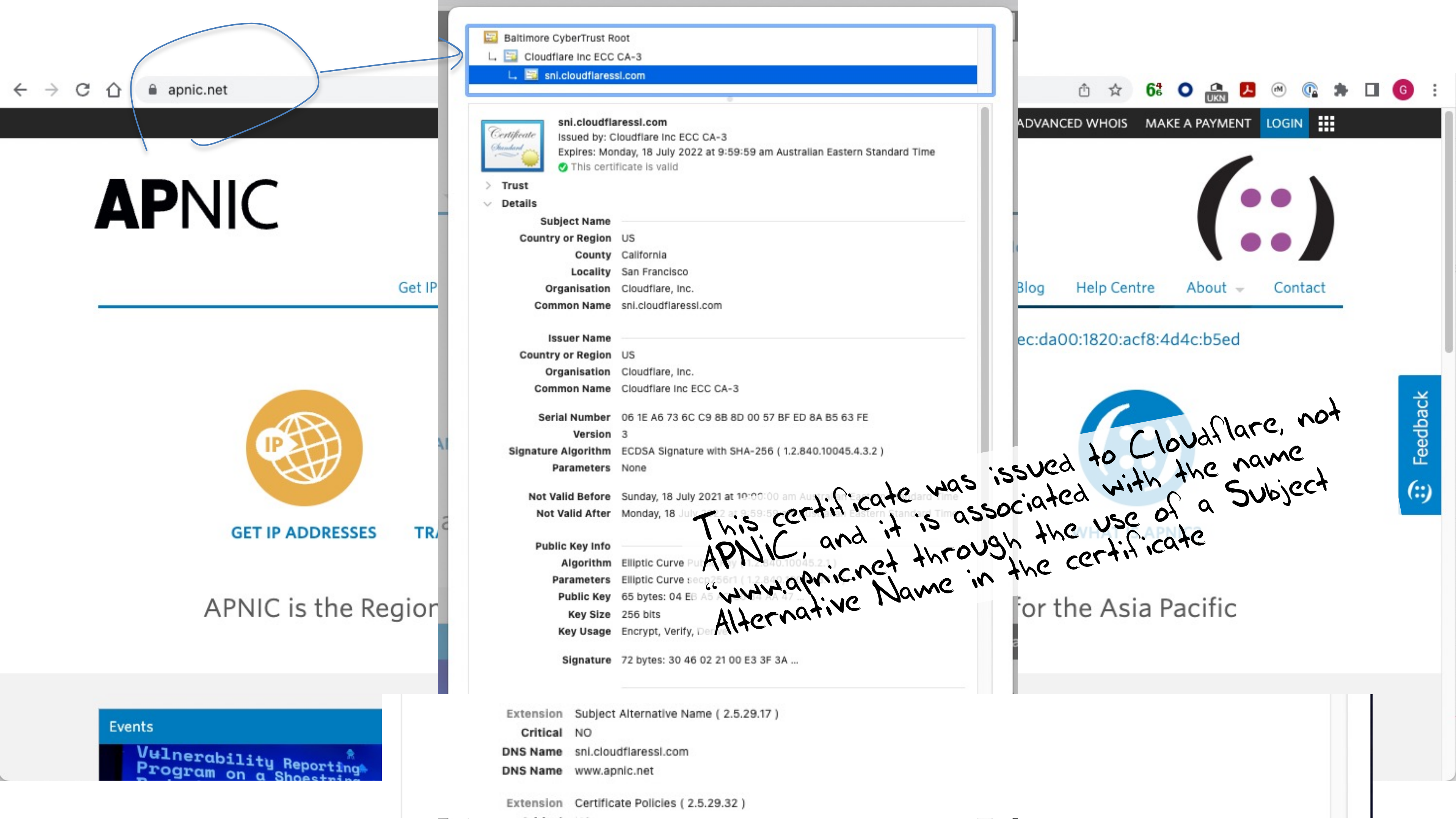
I trust that this is the web site of the Commonwealth Bank because I used the Commonwealth Bank's public key to set up the encrypted connection to the server.

And I can trust that this is the Commonwealth Bank's public key because I trust that Entrust has performed a number of checks before issuing a public key certificate for this public key

And another example

- Let's take www.apnic.net and look at that certificate

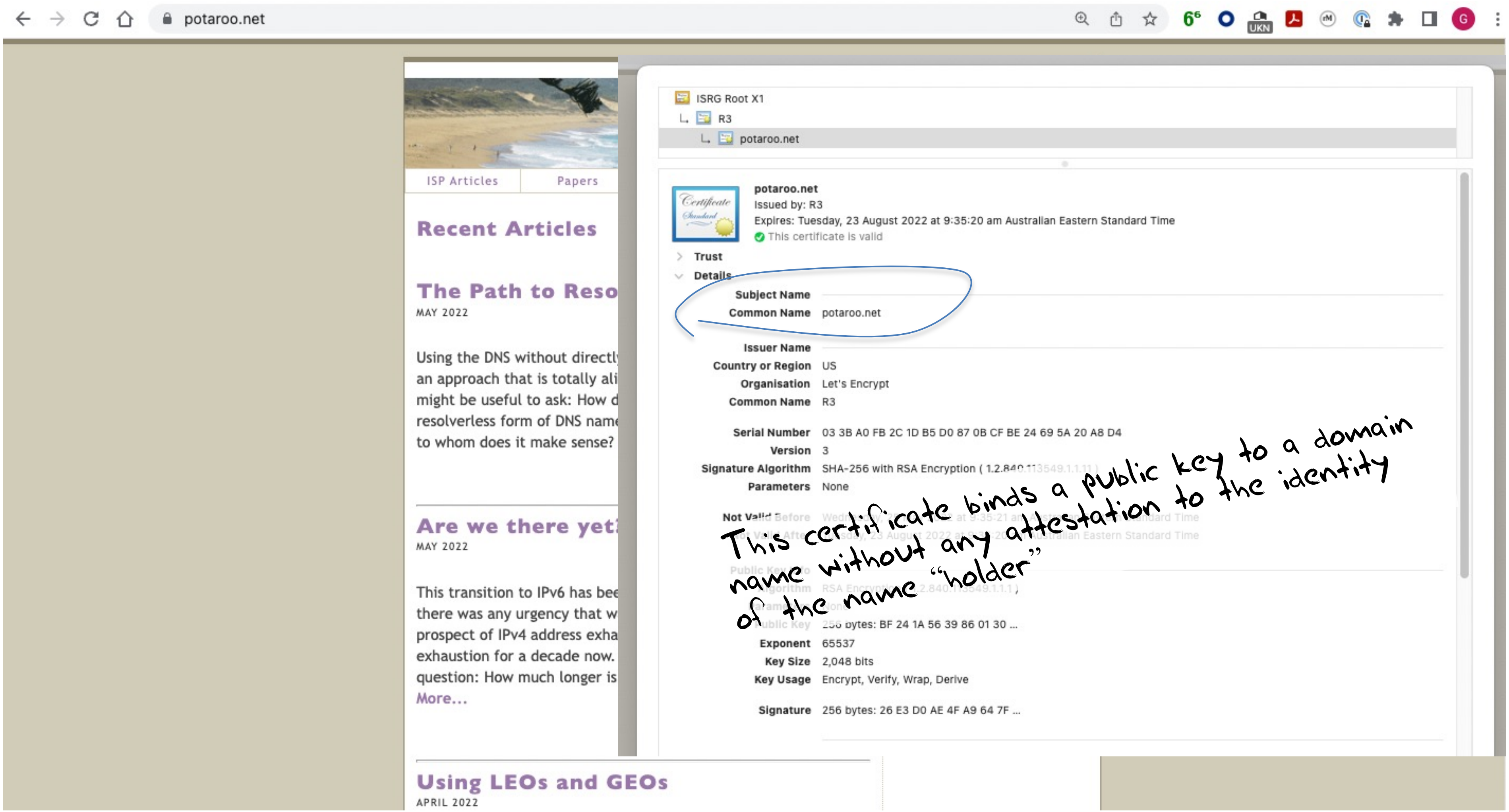

apnic.net
WHOIS & WEBSITE
ADVANCED WHOIS
MAKE A PAYMENT
LOGIN
APNIC
Get IP
Manage IP
Training
Events
Insights
Community
Blog
Help Centre
About
Contact
Your IP address: 2001:8003:1dec:da00:1820:acf8:4d4c:b5ed
IP
GO IPv6
GET IP ADDRESSES
TRANSFER IP ADDRESSES
GO IPv6
PARTICIPATE
WHAT IS APNIC?
Feedback
APNIC is the Regional Internet Registry administering IP addresses for the Asia Pacific
Events
Vulnerability Reporting Program on a Shoestring
Tech matters
Homepage highlight



This certificate was issued to Cloudflare, not APNIC, and it is associated with the name "www.apnic.net through the use of a Subject Alternative Name in the certificate

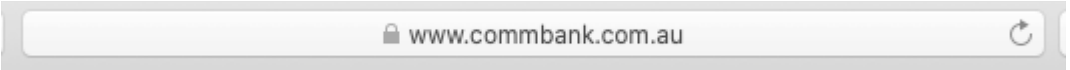
And another

- Let’s look at my own web site, with its certificate issued by Let’s Encrypt

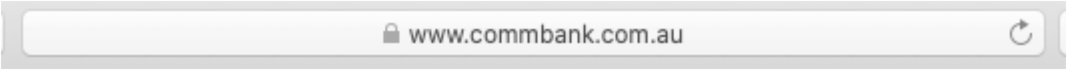


This certificate binds a public key to a domain name without any attestation to the identity of the name "holder"

Spot the Difference

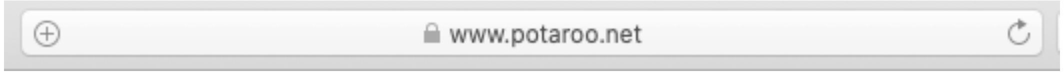


Spot the Difference



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

Spot the Difference

- The certification processes used to issue the certificate were different in each of these cases.
  - One confirmed the identity of the public key holder as well as their association with the domain name
  - The second used a proxy agent and there is **no** association between the entity domain name that is certified here and the proxy agent
  - The third simply associates a public key with a domain name without any form of identification of the holder of the domain name
- They have very different levels of trustworthiness, yet they all display to the user in exactly the same way
  - Because when we tried to differentiate these different levels of trust (such as painting the padlock icon in green) nobody understood what was going on and nobody cared anyway!

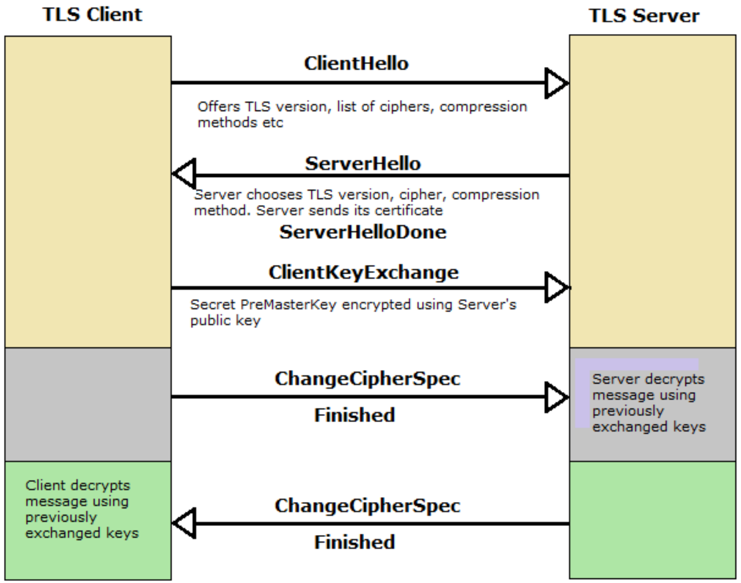
Spot the Difference

- While there are important differences in the trustworthiness in these three certificates, they all display on the user's screen in precisely the same way
- As an attacker, if I can use the lowest threshold of proof to have a counterfeit certificate issued, then perhaps there is a viable attack vector, as the user would not notice the switch to a less strict form of subject identity validation

Moving on…

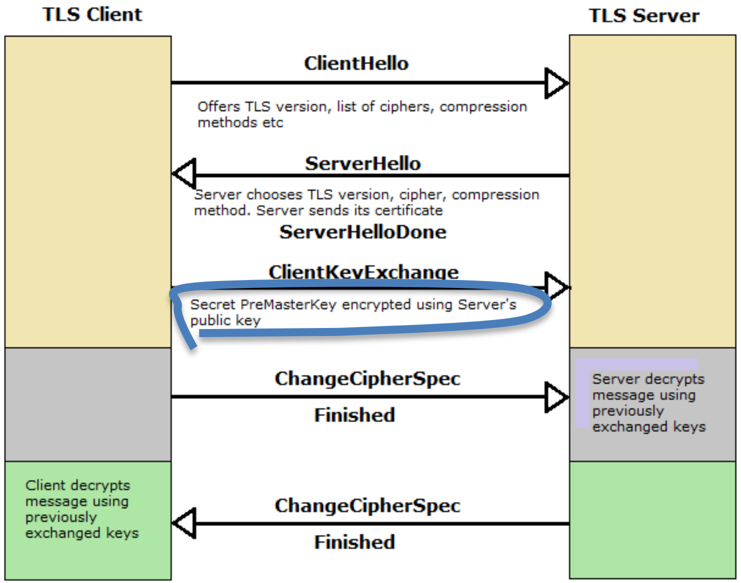
- Ok, so the certificate system is a mess, but the subsequent secure transport session (TLS) still works, right?
- Let's look at the way TLS starts a secure session

Secure Connections using TLS



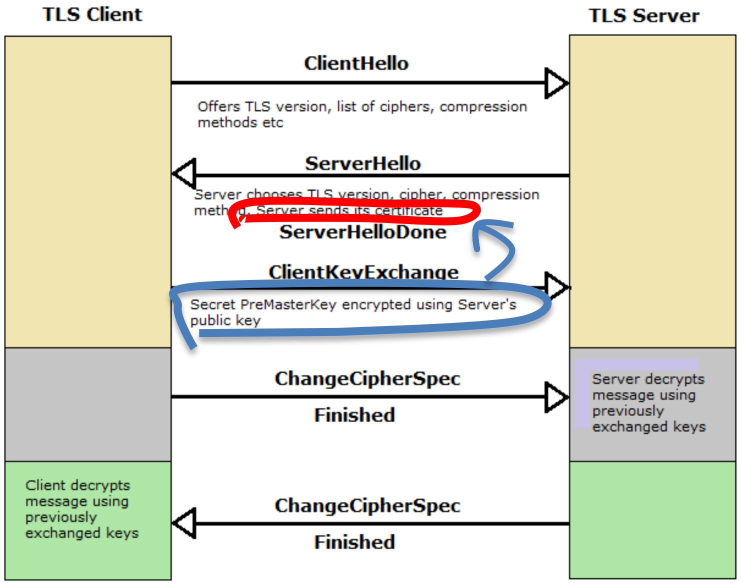
https://rhsecurity.wordpress.com/tag/tls/

Secure Connections using TLS



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

Secure Connections using TLS



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

Entrust Root Certification Authority - G2
└ Entrust Certification Authority - L1M
└ www.commbank.com.au

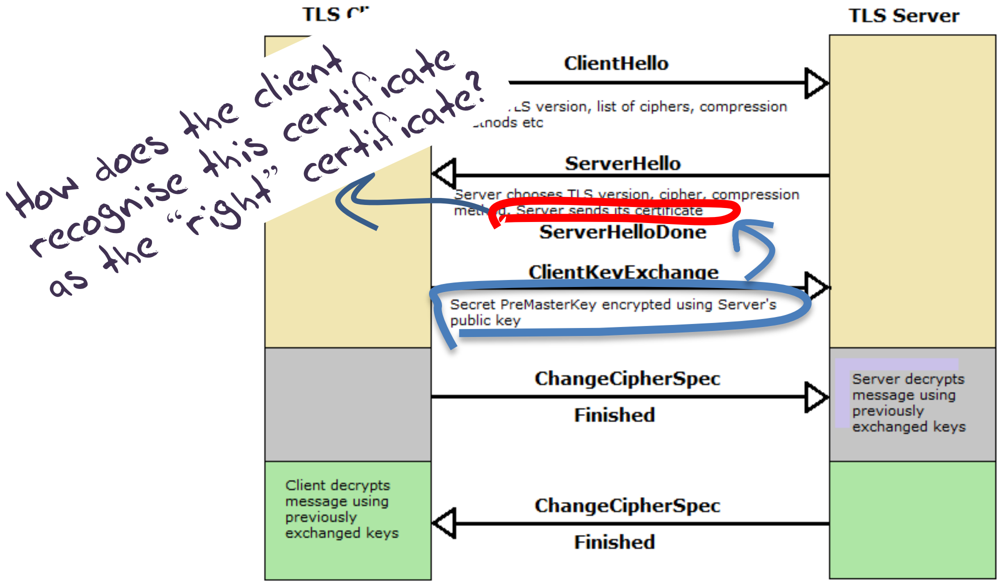
www.commbank.com.au
Issued by: Entrust Certification Authority - L1M
Expires: Saturday, 29 April 2023 at 9:59:12 am Australian Eastern Standard Time
✅ This certificate is valid

› **Trust**

⌄ **Details**

Subject Name	
Country or Region	AU
County	New South Wales
Locality	Sydney
Inc. Country/Region	AU
Organisation	Commonwealth Bank of Australia
Business Category	Private Organization
Serial Number	48 123 123 124
Common Name	www.commbank.com.au
Issuer Name	
Country or Region	US
Organisation	Entrust, Inc.
Organisational Unit	See www.entrust.net/legal-terms
Organisational Unit	(c) 2014 Entrust, Inc. - for authorized use only
Common Name	Entrust Certification Authority - L1M
Serial Number	24 F5 40 B3 F7 9F 29 57 72 A0 F1 1C 6F 3D E7 AB
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters	None
Not Valid Before	Wednesday, 30 March 2022 at 10:59:12 am Australian Eastern Daylight Time
Not Valid After	Saturday, 29 April 2023 at 9:59:12 am Australian Eastern Standard Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	None
Public Key	256 bytes: BF 7E 21 BA 6C E0 A1 9D ...
Exponent	65537
Key Size	2,048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes: C3 28 89 A4 13 51 B0 8A ...

Secure Connections using TLS



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

Entrust Root Certification Authority - G2
Entrust Certification Authority - L1M
www.commbank.com.au

www.commbank.com.au
Issued by: Entrust Certification Authority - L1M
Expires: Saturday, 29 April 2023 at 9:59:12 am Australian Eastern Standard Time
This certificate is valid

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

Trust
Details

Subject Name	
Country or Region	AU
County	New South Wales
Locality	Sydney
Inc. Country/Region	AU
Organisation	Commonwealth Bank of Australia
Business Category	Private Organization
Serial Number	48 123 123 124
Common Name	www.commbank.com.au
Issuer Name	
Country or Region	US
Organisation	Entrust, Inc.
Organisational Unit	See www.entrust.net/legal-terms
Organisational Unit	(c) 2014 Entrust, Inc. - for authorized use only
Common Name	Entrust Certification Authority - L1M
Serial Number	24 F5 40 B3 F7 9F 29 57 72 A0 F1 1C 6F 3D E7 AB
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters	None
Not Valid Before	Wednesday, 30 March 2022 at 10:59:12 am Australian Eastern Daylight Time
Not Valid After	Saturday, 29 April 2023 at 9:59:12 am Australian Eastern Standard Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	None
Public Key	256 bytes: BF 7E 21 BA 6C E0 A1 9D ...
Exponent	65537
Key Size	2,048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes: C3 28 89 A4 13 51 B0 8A ...

Domain Name Certification

- The Commonwealth Bank of Australia has generated a Public/Private key pair
- And they passed a certificate signing request to a company called “Entrust” 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 Entrust and the certificates that they issue
- And I’m prepared to trust them because Entrust NEVER lie!

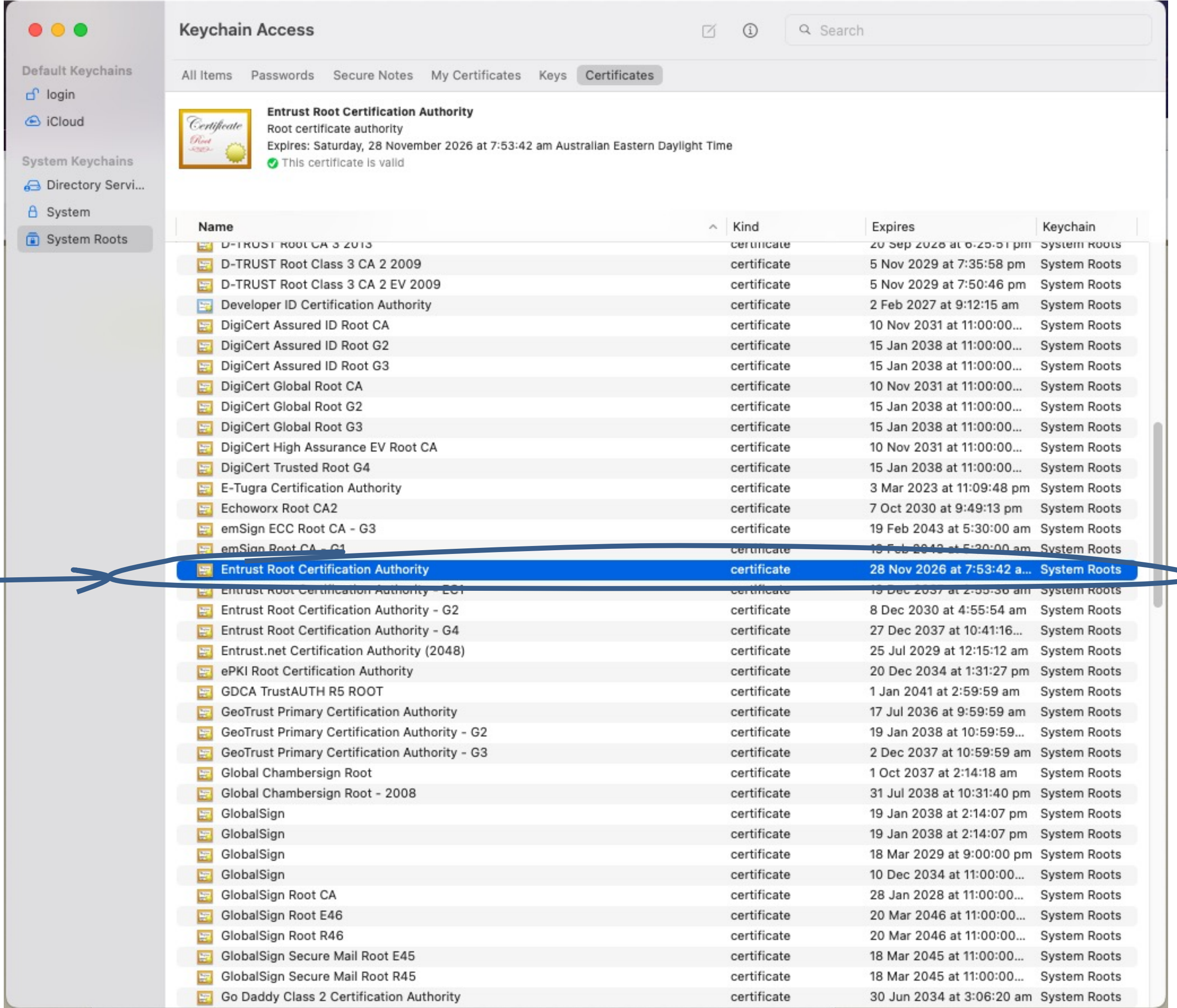
Domain Name Certification

- The Commonwealth Bank of Australia has generated a Public/Private key pair
- And they passed a certificate signing request to a company called "Entrust" 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 Entrust and the certificates that they issue
- And I'm prepared to trust them because Entrust NEVER lie!

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

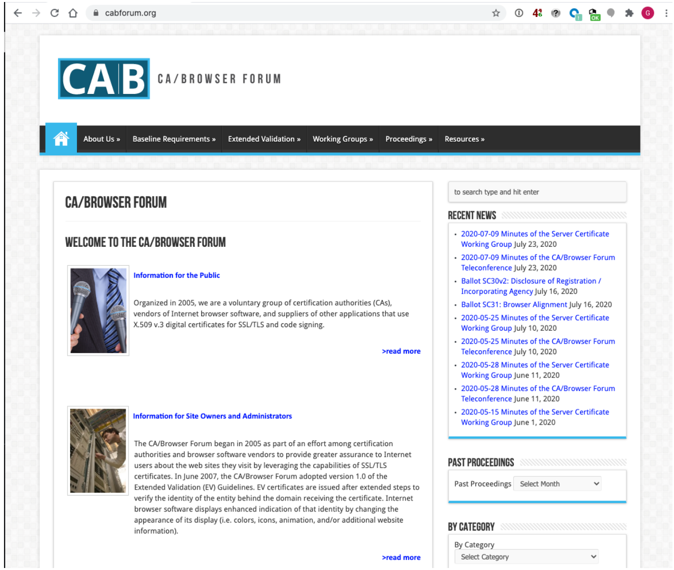
Local Trust

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

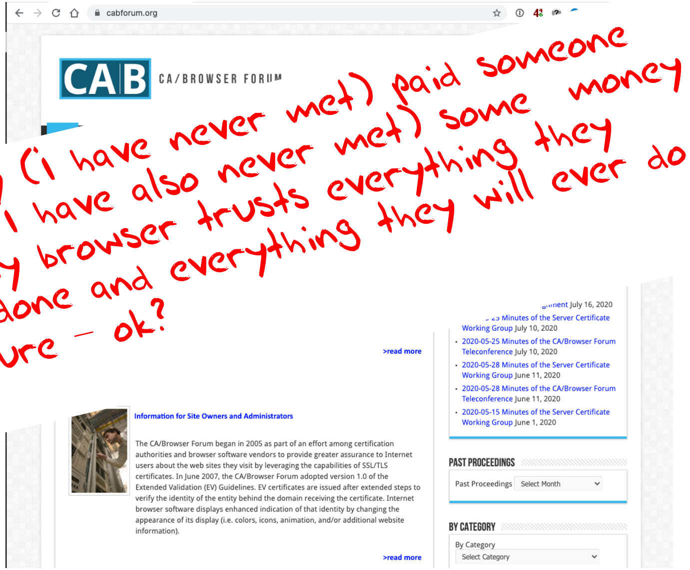
These Certificate Authorities are listed in my computer's trust set because they claim to operate according to the practices defined by the CAB industry forum (of which they are a member) and they **never** lie!



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!

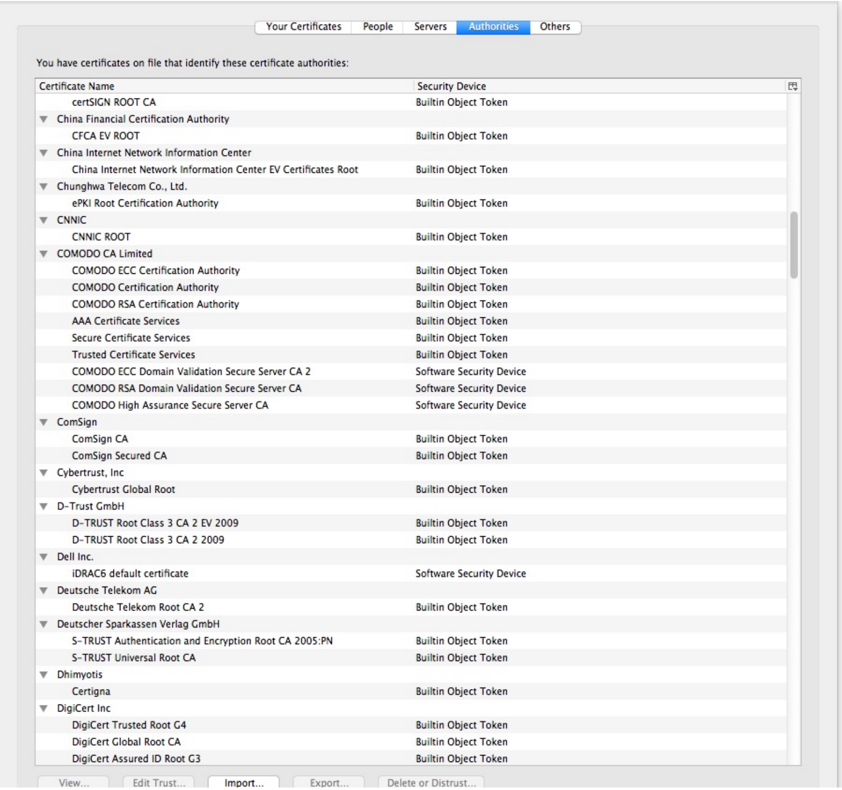
So somebody (I have never met) paid someone else (whom I have also never met) some money and then my browser trusts everything they have ever done and everything they will ever do in the future – ok?



Local Trust or Local Credulity*?

Wow!

Are they **all** trustable?



Certificate Name	Security Device
certSIGN ROOT CA	Builtin Object Token
China Financial Certification Authority	
CFCA EV ROOT	Builtin Object Token
China Internet Network Information Center	
China Internet Network Information Center EV Certificates Root	Builtin Object Token
Chunghwa Telecom Co., Ltd.	
ePKI Root Certification Authority	Builtin Object Token
CNNIC	
CNNIC ROOT	Builtin Object Token
COMODO CA Limited	
COMODO ECC Certification Authority	Builtin Object Token
COMODO Certification Authority	Builtin Object Token
COMODO RSA Certification Authority	Builtin Object Token
AAA Certificate Services	Builtin Object Token
Secure Certificate Services	Builtin Object Token
Trusted Certificate Services	Builtin Object Token
COMODO ECC Domain Validation Secure Server CA 2	Software Security Device
COMODO RSA Domain Validation Secure Server CA	Software Security Device
COMODO High Assurance Secure Server CA	Software Security Device
ComSign	
ComSign CA	Builtin Object Token
ComSign Secured CA	Builtin Object Token
Cybertrust, Inc	
Cybertrust Global Root	Builtin Object Token
D-Trust GmbH	
D-TRUST Root Class 3 CA 2 EV 2009	Builtin Object Token
D-TRUST Root Class 3 CA 2 2009	Builtin Object Token
Dell Inc.	
iDRAC6 default certificate	Software Security Device
Deutsche Telekom AG	
Deutsche Telekom Root CA 2	Builtin Object Token
Deutscher Sparkassen Verlag GmbH	
S-TRUST Authentication and Encryption Root CA 2005:PN	Builtin Object Token
S-TRUST Universal Root CA	Builtin Object Token
Dhimyotis	
Certigna	Builtin Object Token
DigiCert Inc	
DigiCert Trusted Root G4	Builtin Object Token
DigiCert Global Root CA	Builtin Object Token
DigiCert Assured ID Root G3	Builtin Object Token

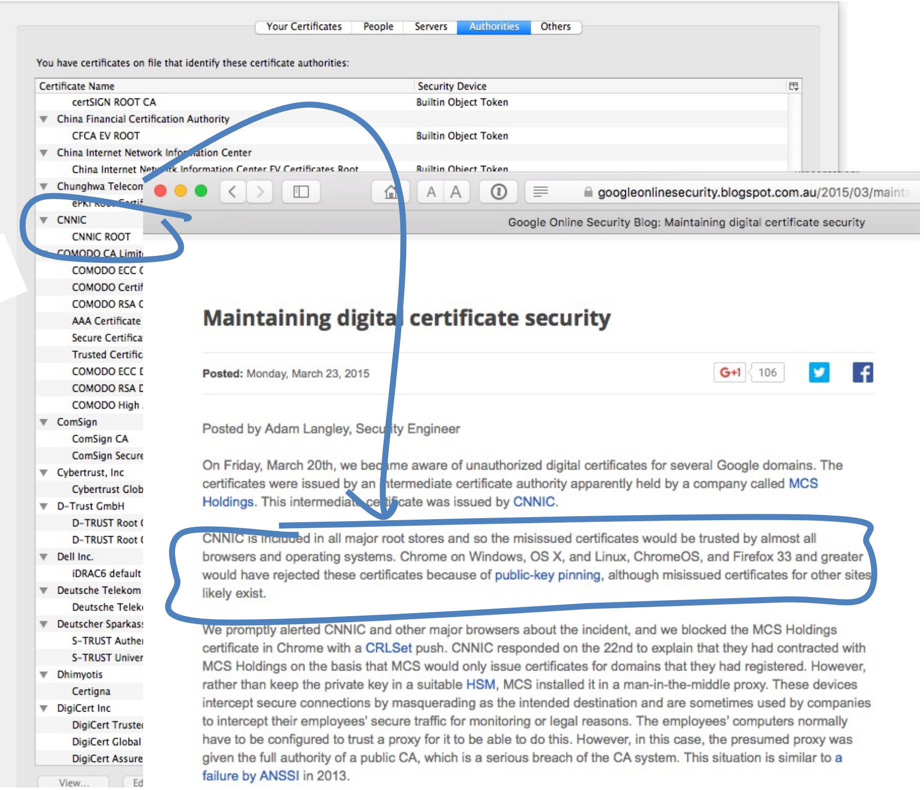
* cre·du·li·ty
/krəˈd(y)o͞olədē/
noun
a tendency to be too ready to believe that something is real or true.

Local Credulity

Wow!

Are they **all** trustable?

Evidently Not!

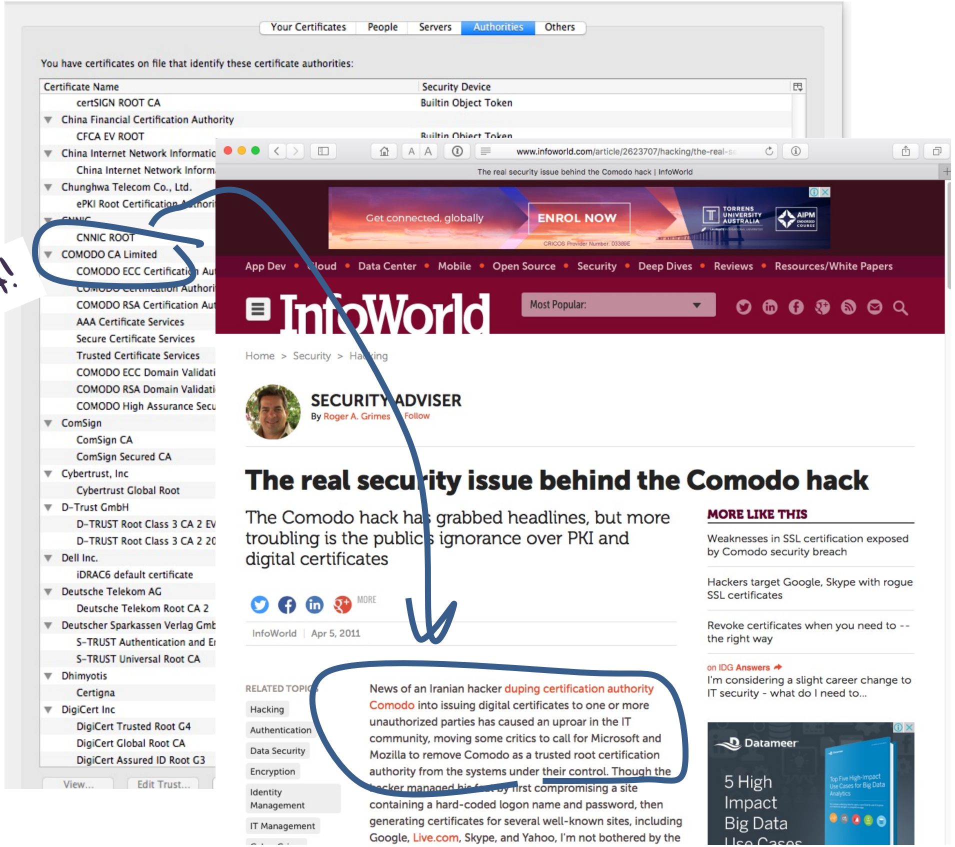


Local Credulity

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=A8F14F52CC1282D7153A13316E7DA39E6AE37B1A10C16288B9024A9B9DC3C4C6

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

security.googleblog.com/2018/03/distrust-of-symantec-p

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

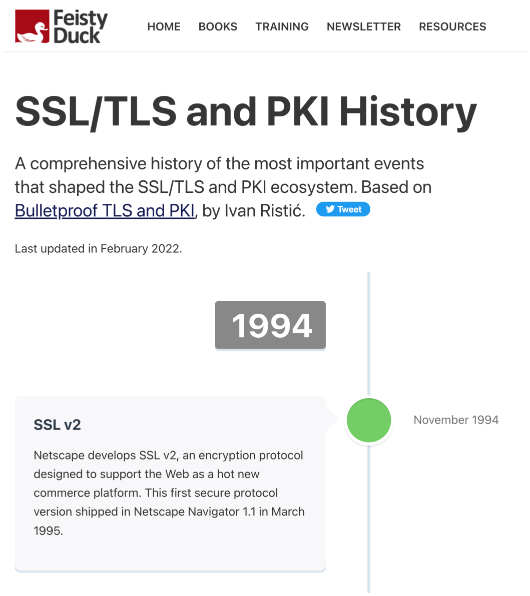
If your site is using a SSL/TLS certificate from Symantec that was issued before June 1, 2016, it will stop functioning in Chrome 66, which could already be impacting your users.

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.

These are isolated events

No, they're not:

https://www.feistyduck.com/ssl-tls-and-pki-history/



With unpleasant consequences when it all goes wrong

With unpleasant consequences when it all goes wrong

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

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 s
pu
- Th NY
CA validate a certificate!

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!

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 it's all OK

Really.

- Because ‘bad’ certificates can be revoked
- And browsers **always** check revocation status of certificates before they trust them

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

So, we can't count on revocation

- If we can’t revoke certificates, then we need to reduce certificate lifetimes

So, we can't count on revocation

- If we can't revoke certificates then we need to reduce certificate lifetimes
- But we are not doing that!

emSign ECC Root CA - G3	certificate	19 Feb 2043 at 5:30:00 am	System Roots
emSign Root CA - G1	certificate	19 Feb 2043 at 5:30:00 am	System Roots
Entrust Root Certification Authority	certificate	28 Nov 2026 at 7:53:42 a...	System Roots
Entrust Root Certification Authority - EC1	certificate	19 Dec 2037 at 2:55:36 am	System Roots

Yes, 2026!!!

So, we can't count on revocation

- If we can't revoke certificates then we need to reduce certificate lifetimes
- What's a "safe" certificate lifetime?


ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE FORU

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

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.

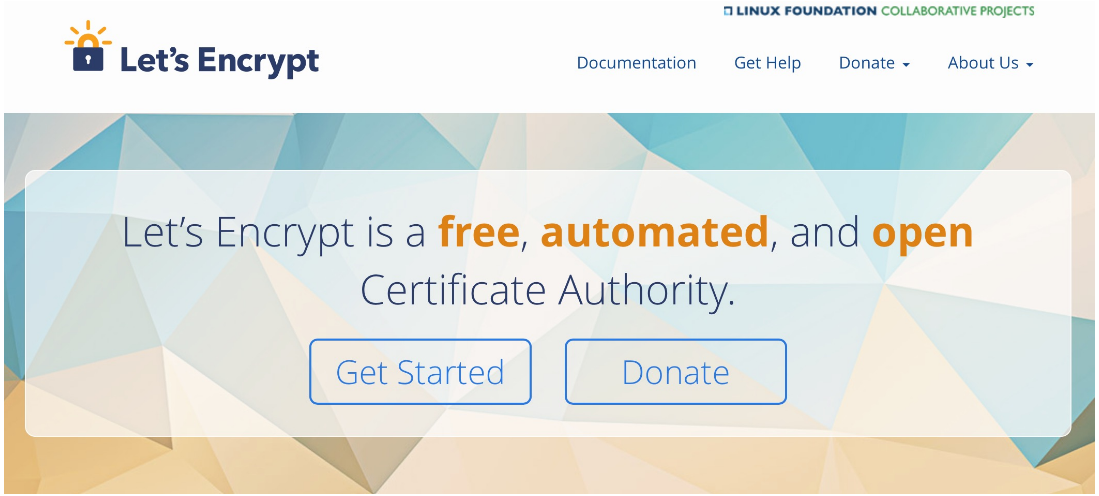
The incident, which started around 6 AM California time, hijacked roughly 1,300 IP addresses, Oracle-owned Internet Intelligence said on Twitter. The malicious redirection was caused by fraudulent routes that were announced by Columbus, Ohio-based eNet, a large Internet service provider that is referred 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

So, we can't count on revocation

- If we can’t revoke certificates then we need to reduce certificate lifetimes
- What’s a “safe” certificate lifetime?
- If we want 2 hours or less, then we need to think hard about how to achieve this

How can we make certificates better?

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



How can we make certificates better?

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

How can we make certificates better?

Option B: White Listing and Pinning with HSTS

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

transport_security_state_static.json

```
// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use 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.

// 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 make certificates better?

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

```
// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use 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.

// 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 make certificates better?

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

Option B: White Listing and Pinning with HSTS

its just Google protecting itself and no one

https://code.google.com/p/chromium/codesearch#chromium/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

```
//       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 make certificates better?

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	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

How can we make certificates better?

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.

This is true

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

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.

This is a fail

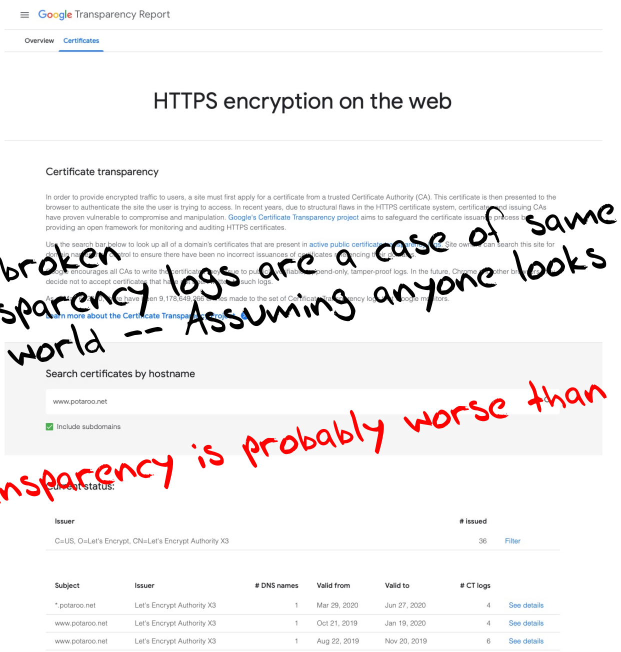
Current status:

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

How can we maske certificates better?

Option C: Certificate Transparency



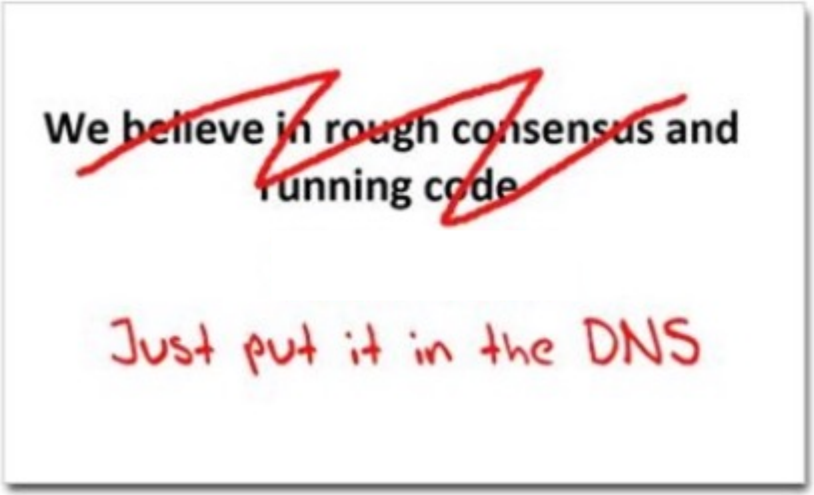
It's just so broken

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 make certificates better?

Option D: Use the DNS!



www.cafepress.com/nxdomain

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 HST[illegible]?
- Why not query the DNS fo[illegible]er CA?
- Why not query th[illegible]e hash of the domain name cert?
- Why not [illegible] 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) P. Hoffman
Request for Comments: 6698 VPN Consortium
Category: Standards Track J. Schlyter
ISSN: 2070-1721 Kirei AB
Augu...

The DNS-Based Authentication of Nam...
Transport Layer Securit...

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... ...trators 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.

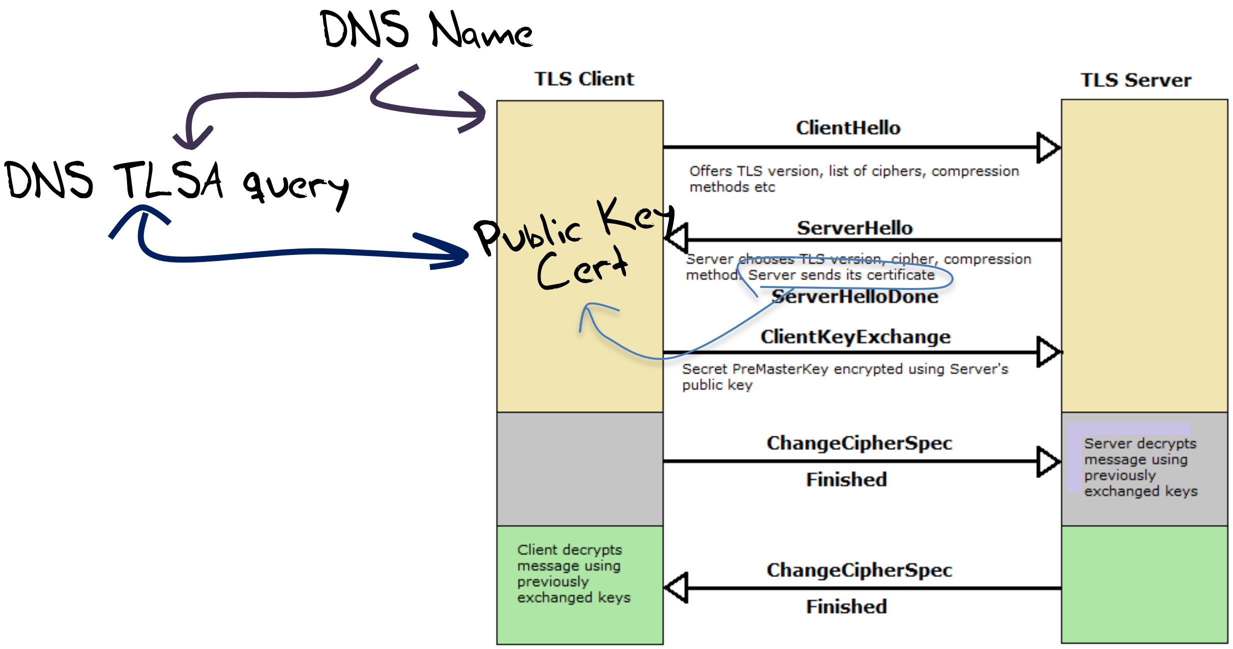
Status of This Memo

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 yo have an unbroken signature chain to the roo
- At this point you ca the TLSA record as the authentic record, and set up a TLS session based on this data

Yes, but No!

DANE + DNSSEC

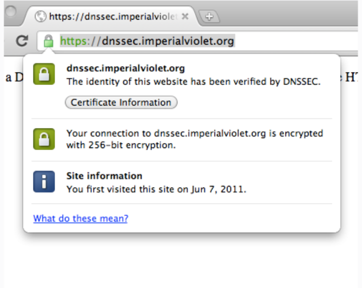
ImperialViolet

DNSSEC authenticated HTTPS in Chrome (16 Jun 2011)

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

DANE validation can be SO SLOW!

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.

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 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 by moving on from X.509 public key certificates:

- 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 ioh ... oving on from X.509 public ke...

But nothing has happened for more than a decade!

Why not?

U

U ... TLS Key Exchange

St ... ESA + sig bundle into TLS

2. Looking Forward

Why is this so hard?

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 conversations of their citizens
- Others want to exploit weaknesses in the system to leverage a competitive advantage
- Some people think users prefer faster applications, even if they have security 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 someone 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!

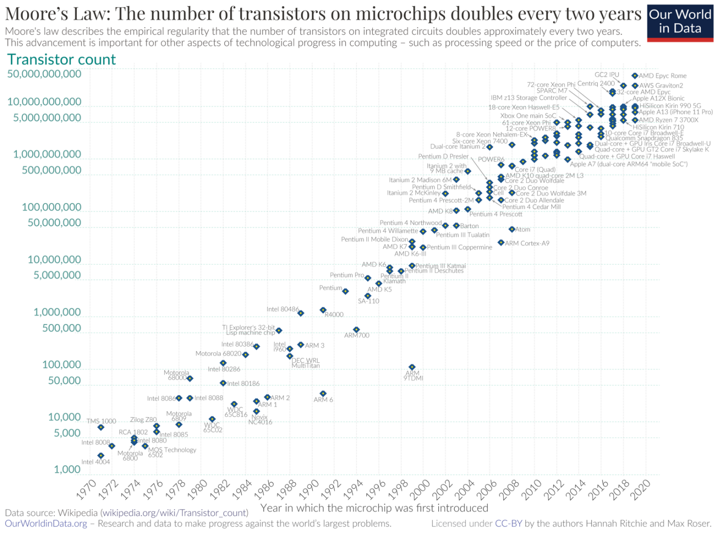
Will it get harder?

Quantum Risk

- Public/Private key cryptography does not create “impossible” problems. It uses “hard” problems.
- But “hard depends on the speed of available CPUs and the way that CPUs calculate
- Quantum computing can potentially solve such “hard” problems, assuming that we get to the point of being able to operate quantum computers with large numbers of qbits.
- So far that challenge has proved elusive, but a lot of effort is being invested into quantum computing these days
- When it happens, we will need to turn to more complex crypto algorithms that have larger keys and require more compute power to use

Will it get more expensive?

- So far Moore's Law has absorbed the incremental cost of crypto
- As we get to 3nm tracks on chips further reductions in size and unit cost are proving to be a major challenge
- Which implies that robust crypto may become more expensive to use
- Who is going to pay the incremental cost of highly robust crypto?



Silicon Chip transistor counts

It's a tough problem…

A rather bleak prognosis from the Economist – don't look for technology to improve this rather disturbing situation!

They suggest looking at economics and markets to try and address this problem

The problem with this suggestion is that there is no natural market that provides incentive for highly robust and secure technologies. The major market incentives are based on driving down unit costs of service delivery, and security is an obvious point of avoidable cost

The Economics of Security

- Effective security for services and infrastructure is a market failure in the IT industry
- Consumers are unwilling to pay a major price premium for a highly robust service
- Service providers do not have any market-based incentive to add robust security to their products and offerings
- The reason why the public sector is undertaking investment in cyber defence measures is that the private sector is not naturally motivated to do so!

The Economics of Security

- Domain Name certificates have only taken off when the cost of obtaining them has dropped to zero, and the demonstration of proof of control is cursory
- And in a demonstration that Gresham's Law applies equally well in security, the low-quality cheap certificate product has driven out other forms of extended validation certification

The Internet of Things

Only makes it so much worse:....

"The market can't fix this because neither the buyer nor the seller cares.

The owners of the low-cost devices used in distributed denial-of-service attacks don't care. Their devices were cheap to buy, they still work, and they don't know any of the victims of the attacks.

The sellers of those devices don't care: They're now selling newer and better models, and the original buyers only cared about price and features.

There is no market solution, because the insecurity is what economists call an externality: It's an effect of the purchasing decision that affects other people. Think of it as a kind of invisible pollution."

https://www.schneier.com/blog/archives/2017/02/security_and_th.html

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

Is this another of those massive challenges of our time?

We just don’t have the tools to figure out how to stop this environment being fatally overrun by these devices:

- We can’t improve their quality
- We can’t keep building ever larger DOS barriers
- We can’t regulate behaviours of the equipment, their makers or distributors

What a dysfunctional mess we've created!

3. Vague Glimmers of Hope

This is not a new problem

- We had a similar problem in the aviation industry
  - Safety was a real issue for the industry
  - The response was to shift the emphasis in investigation of incidents from blame attribution to primary cause identification
- Do we need open disclosure requirements for IT goods and services?

This is not a new problem

- Industry-based safety standards have been used in other industries
- Do consumer products and services need to comply to a set of base standards relating to operational robustness?
- In today’s world of digital goods and services how would such standards be applied?

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.**

Questions?