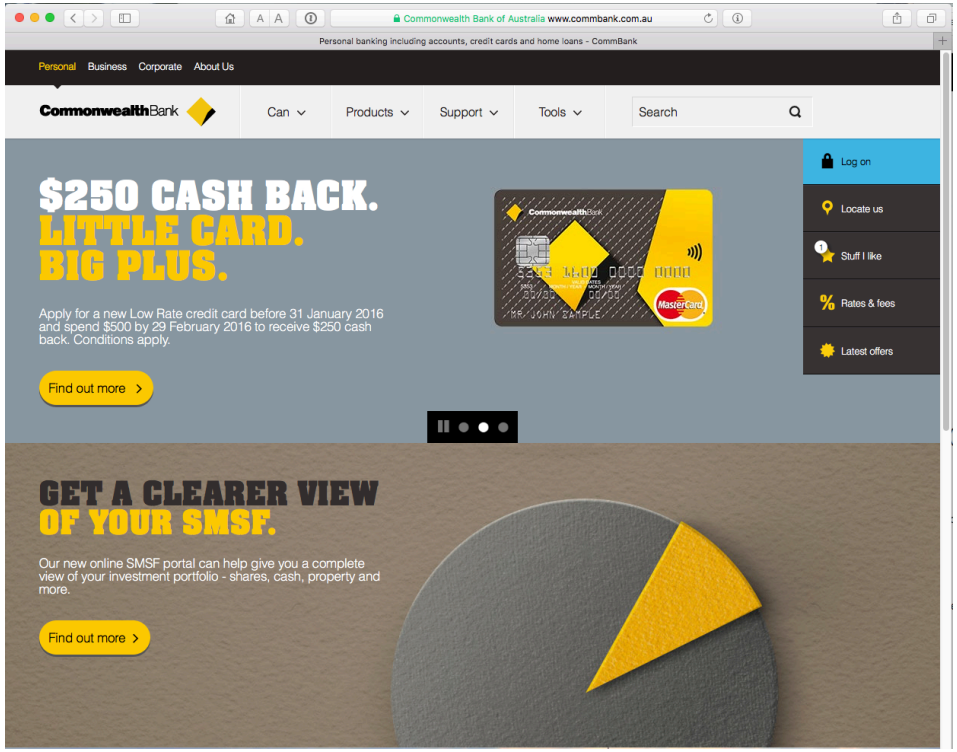
Why Dane?

Geoff Huston
Chief Scientist, APNIC

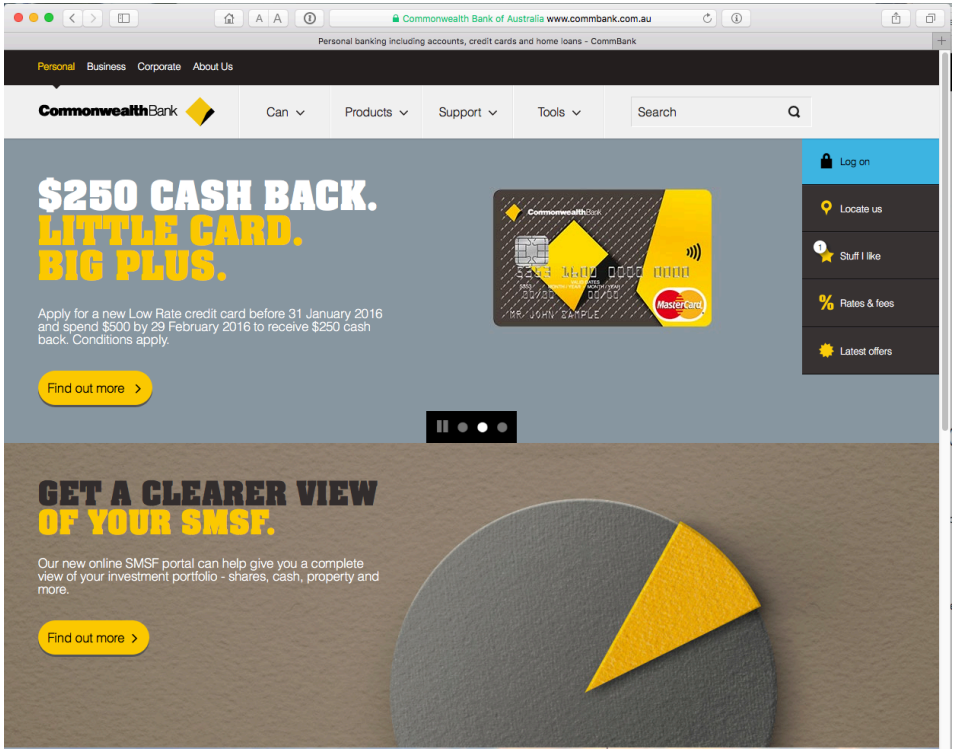
Which Bank? My Bank!



I hope!

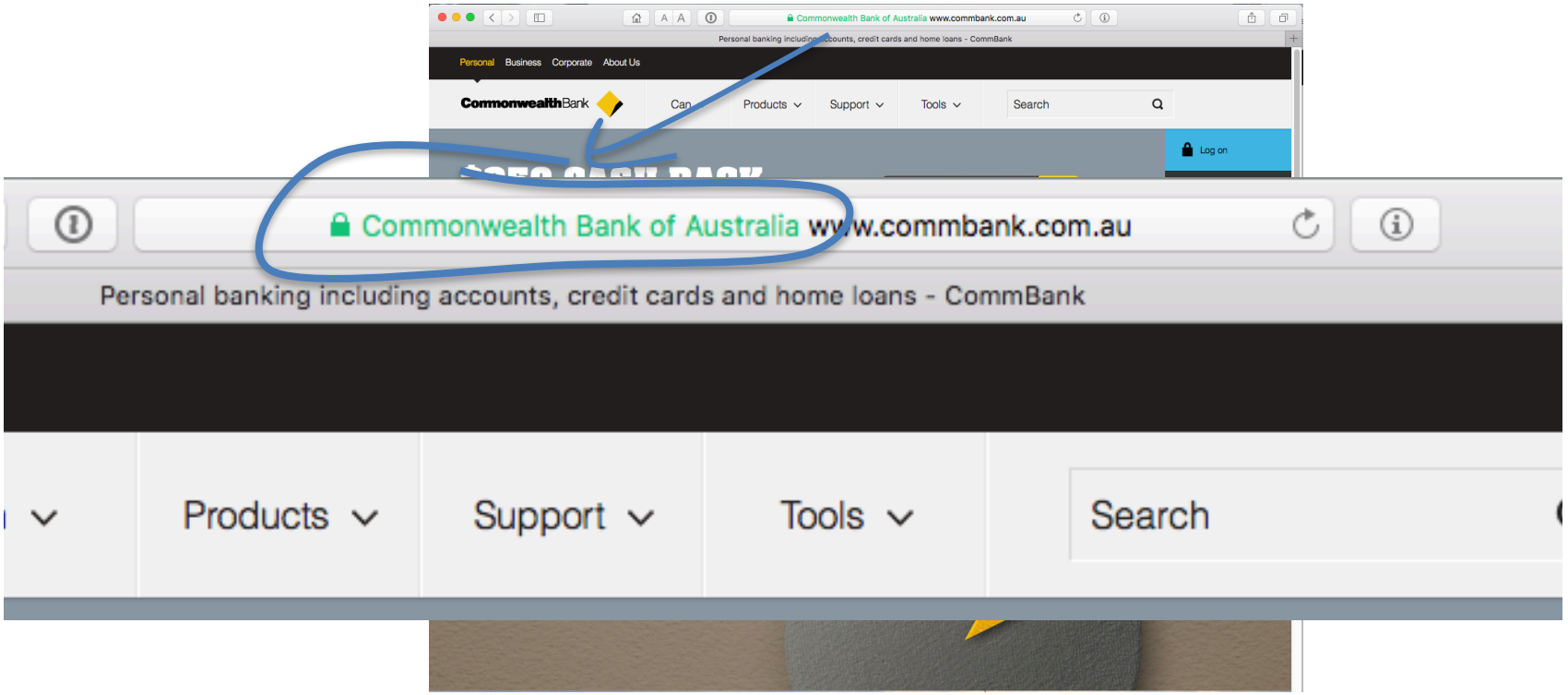
Security on the Internet

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



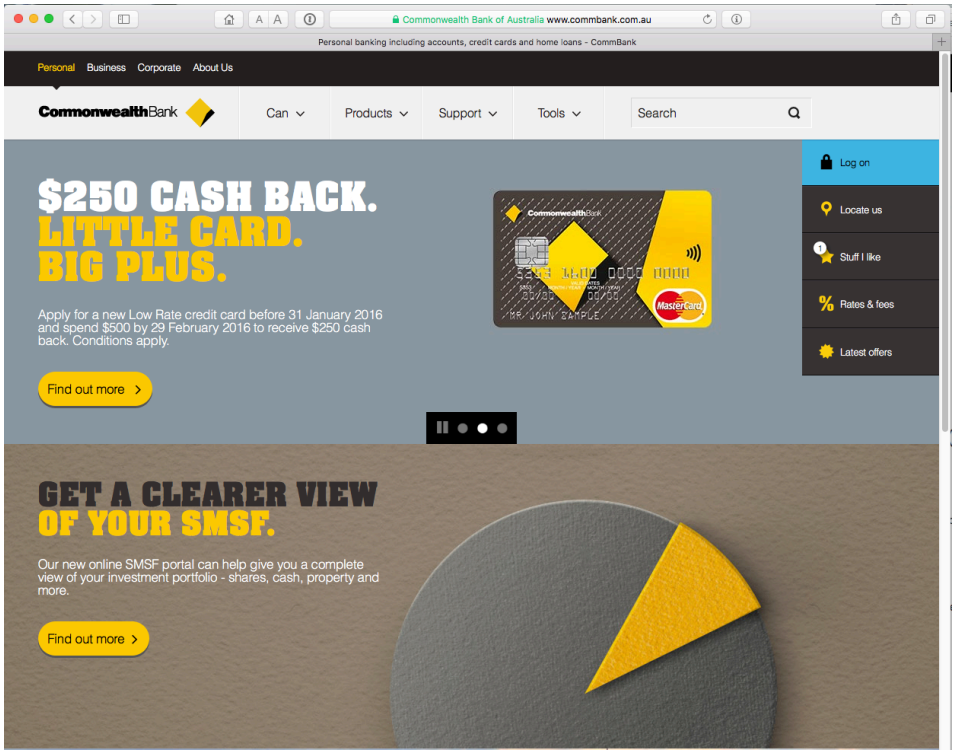
Security on the Internet

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



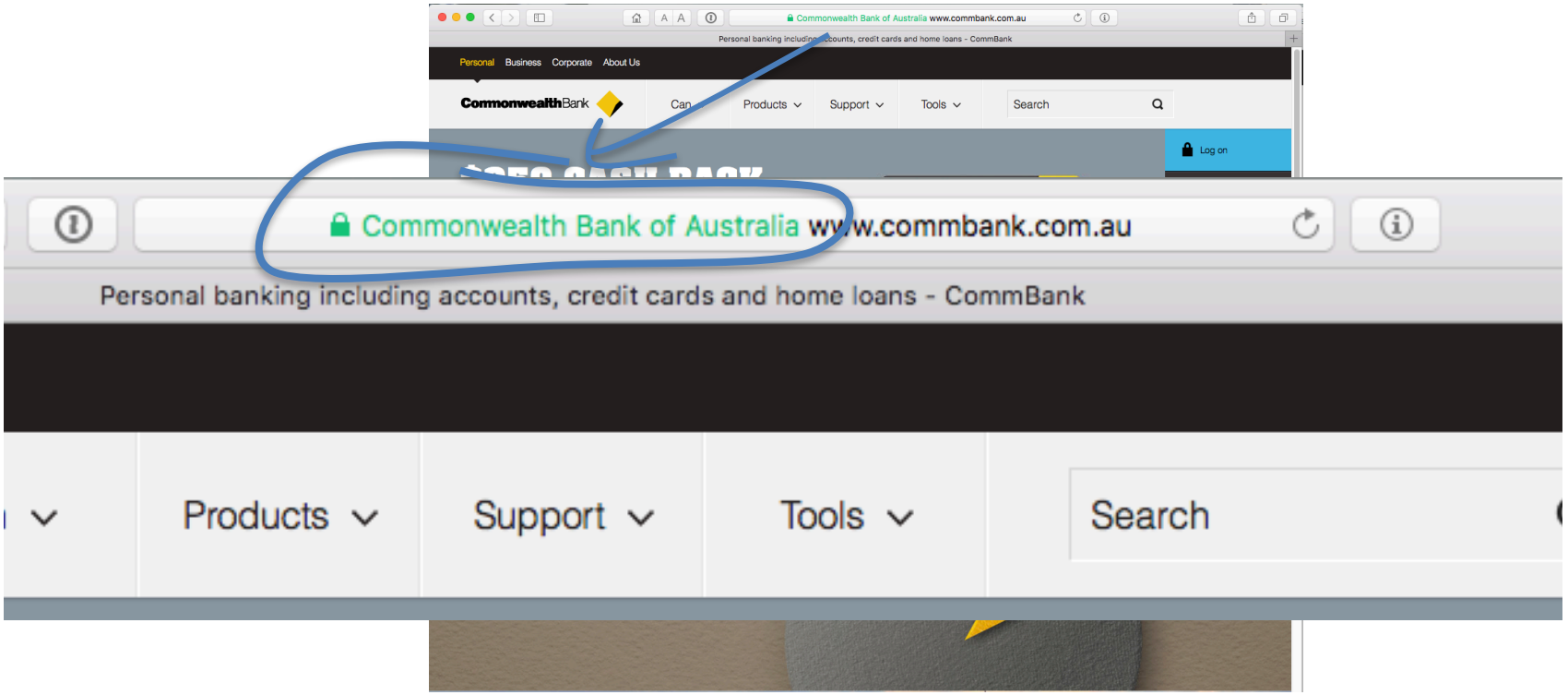
Security on the Internet

Also, how can you keep your session a secret from wire(less) snoopers?



Security on the Internet

Also, how can you keep your session a secret from wire(less) snoopers?



Opening the Connection: First Steps

Client:

DNS Query:

www.commbank.com.au?

DNS Response:

104.97.235.12

TCP Session:

TCP Connect 104.97.235.12, port 443

Hang on…

```
$ dig -x 104.97.235.12 +short
a104-97-235-12.deploy.static.akamaitechnologies.com.
```

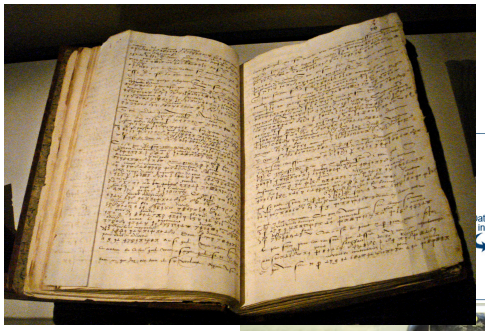
That's **not** an IP addresses that was allocated to the Commonwealth Bank!

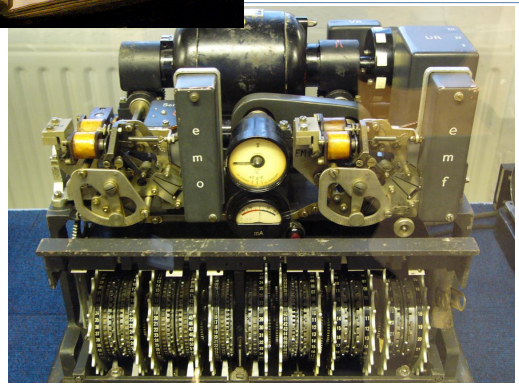
> The Commonwealth Bank of Australia has 140.168.0.0 - 140.168.255.255 and 203.17.185.0 - 203.17.185.255

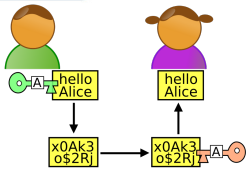
So why should my browser trust that 104.97.235.12 is really the "proper" web site for the Commonwealth Bank of Australia, and not some dastardly evil scam designed to steal my passwords and my money?

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

Its all about cryptography







The Basic Challenge

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

The Power of Primes

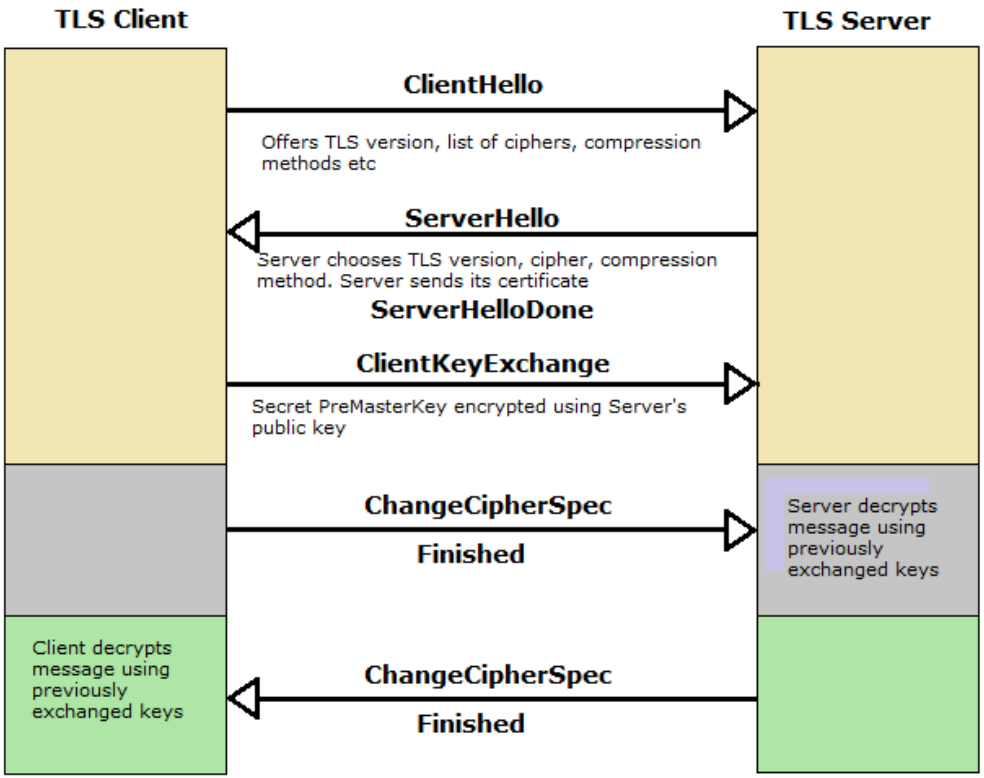
$$(m^e)^d \equiv m \pmod{n}$$

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?

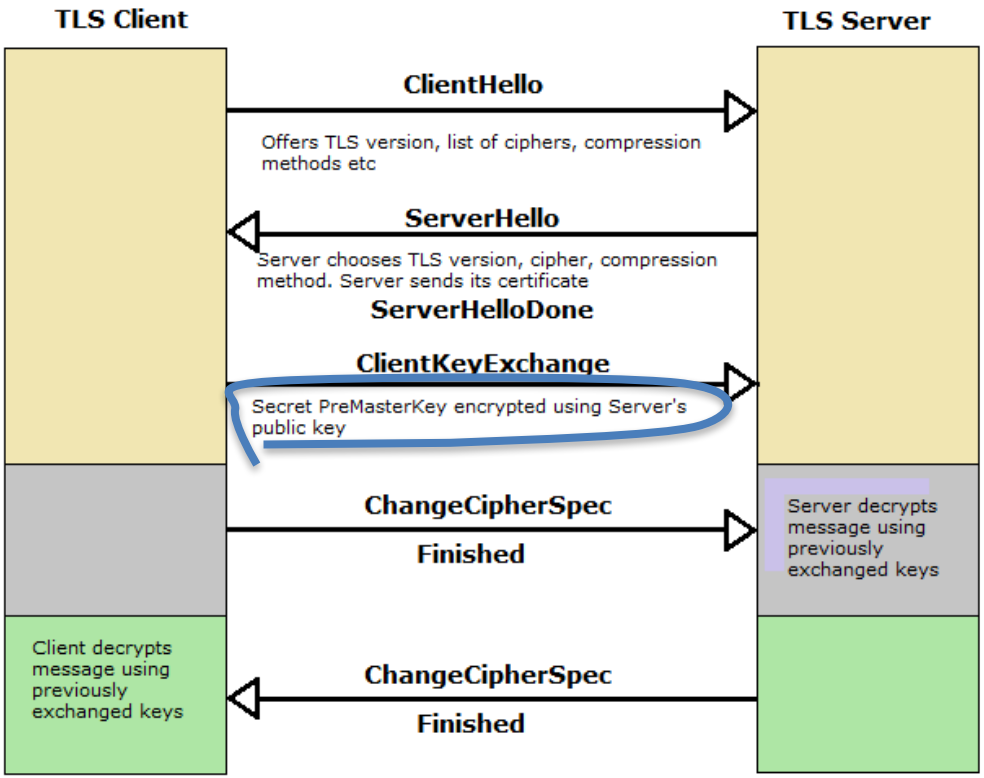
Because much of the foundation of internet Security rests upon this prime number relationship

TLS Connections



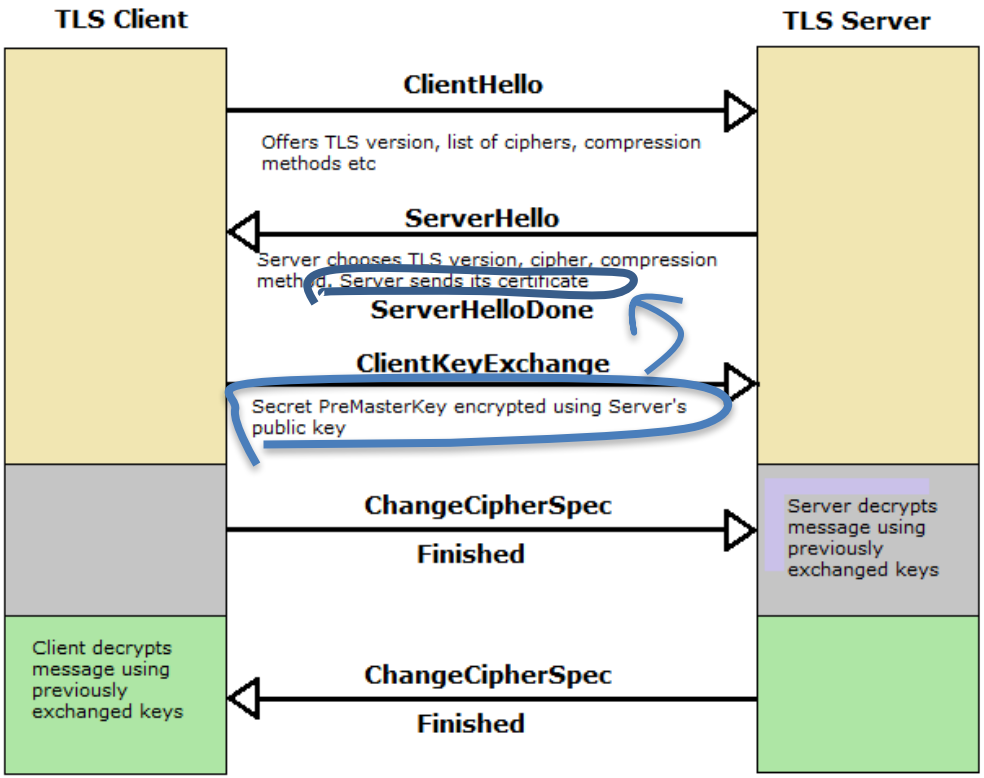
https://rhsecurity.wordpress.com/tag/tls/

TLS Connections



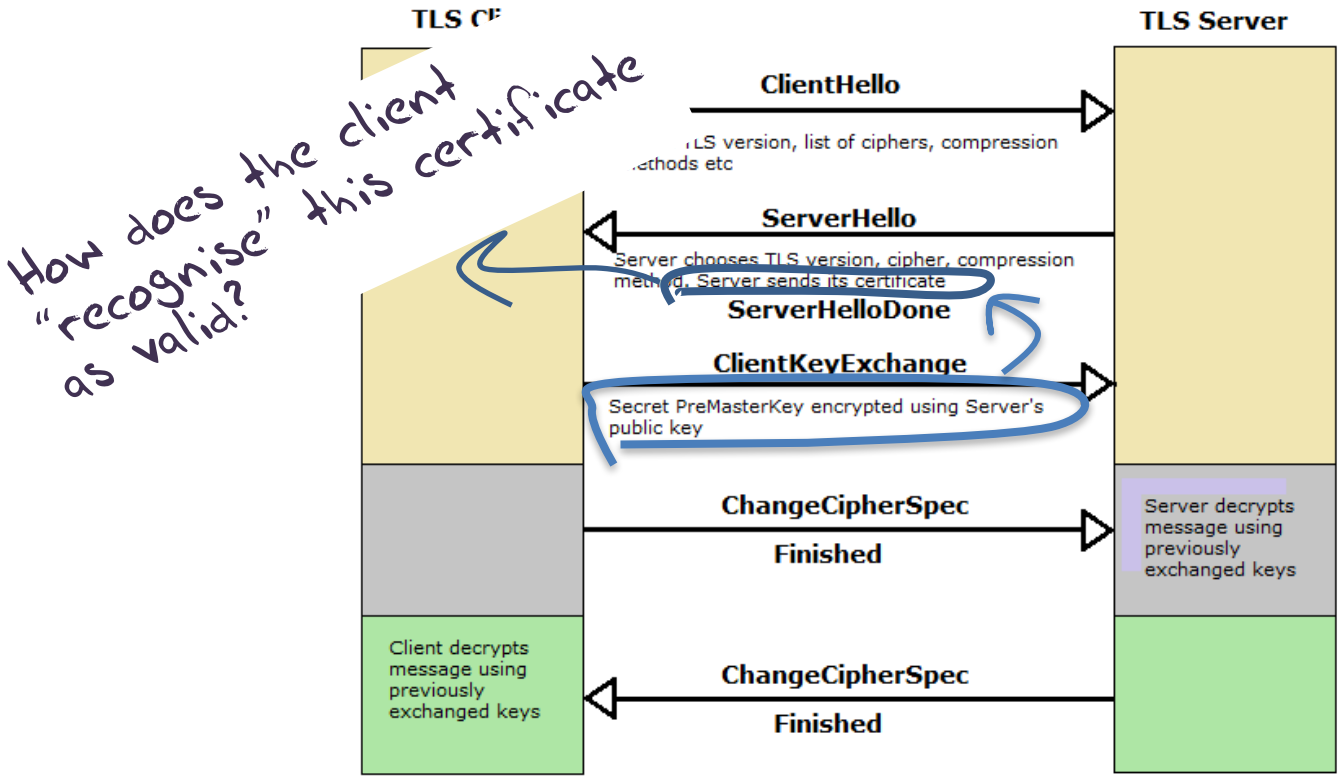
https://rhsecurity.wordpress.com/tag/tls/

TLS Connections

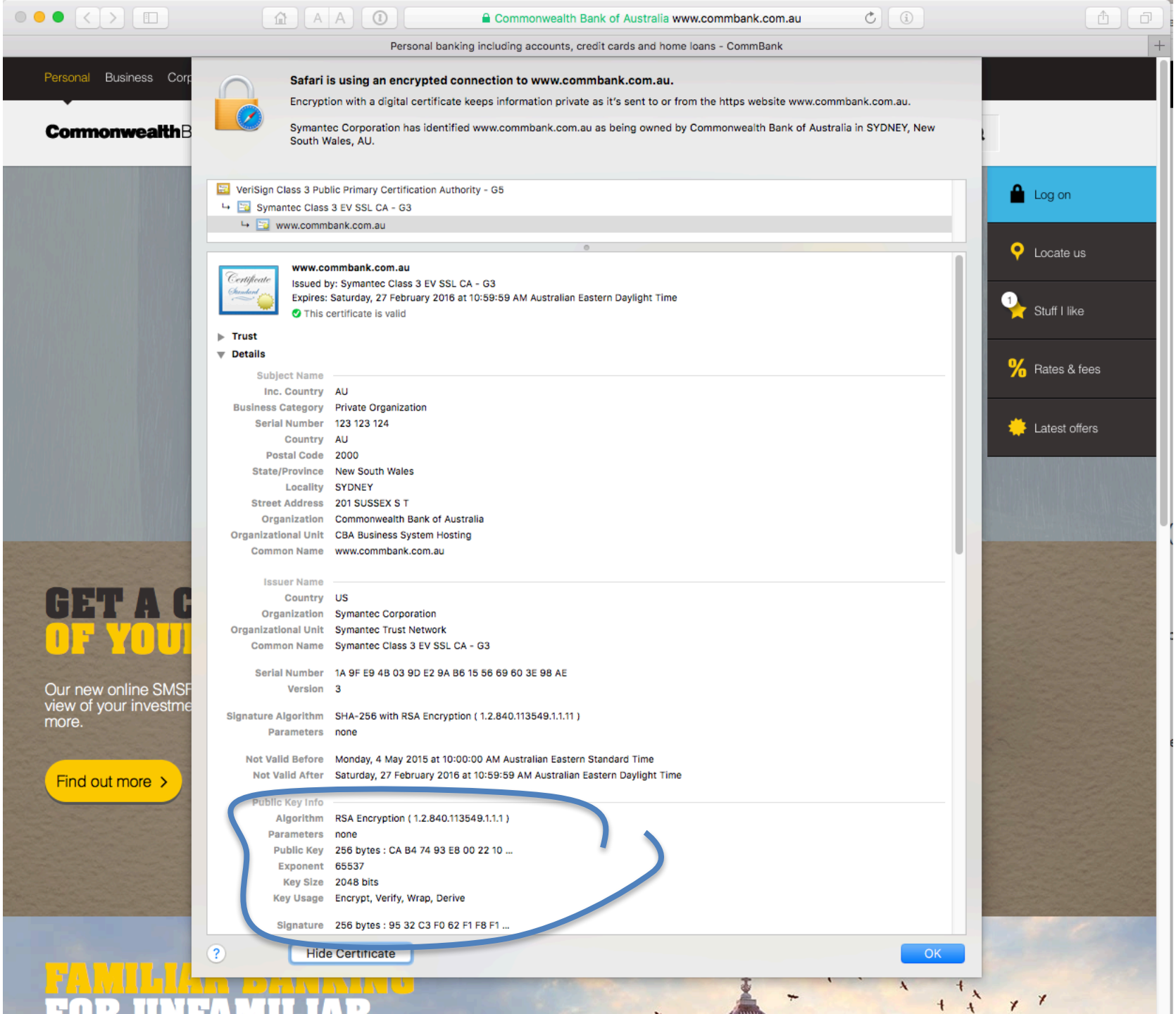


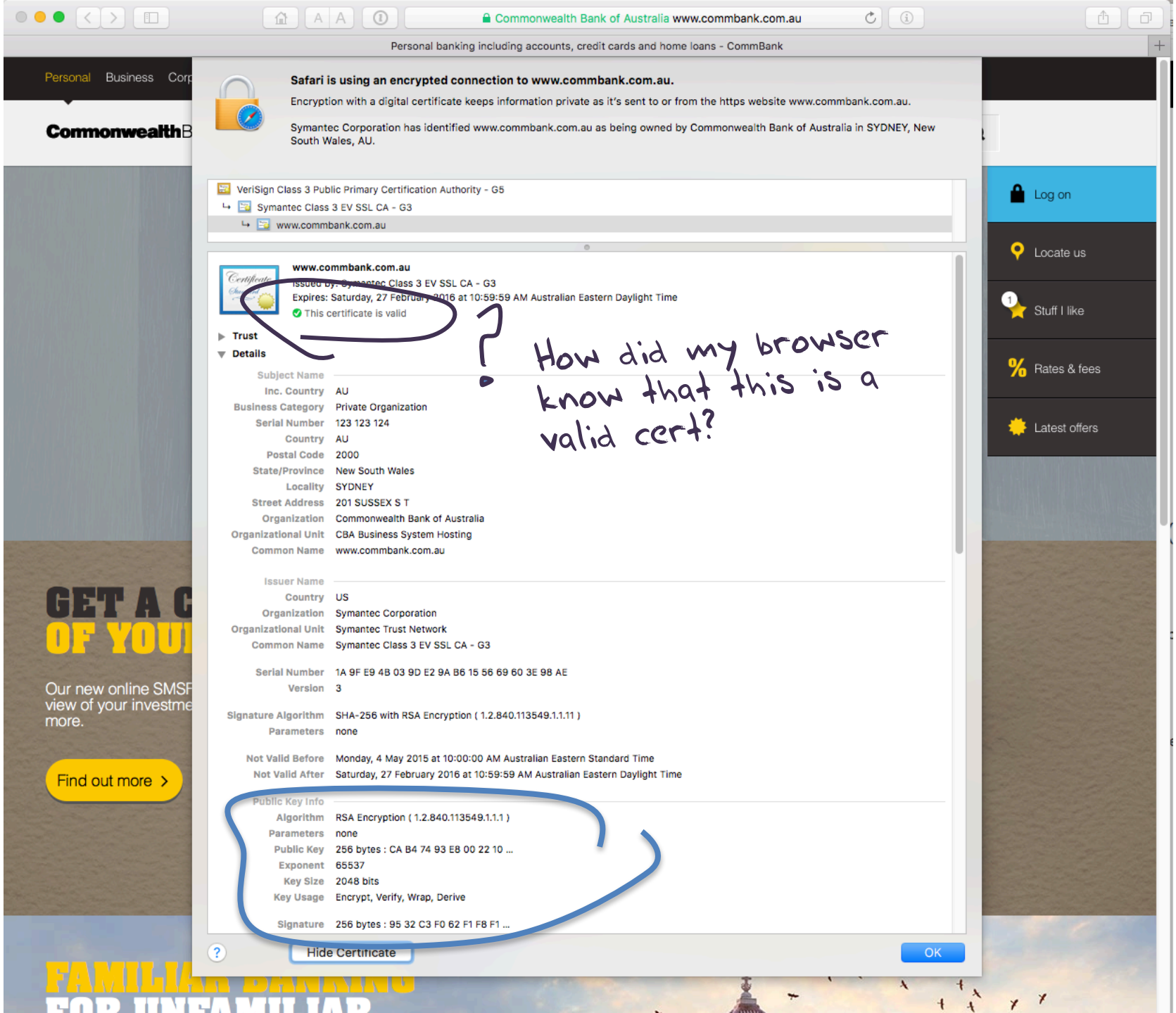
https://rhsecurity.wordpress.com/tag/tls/

TLS Connections



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


Commonwealth Bank of Australia www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank
Personal
Business
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.
Symantec Corporation has identified www.commbank.com.au as being owned by Commonwealth Bank of Australia in SYDNEY, New South Wales, AU.
VeriSign Class 3 Public Primary Certification Authority - G5
Symantec Class 3 EV SSL CA - G3
www.commbank.com.au
www.commbank.com.au
Issued by: Symantec Class 3 EV SSL CA - G3
Expires: Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time
This certificate is valid
Trust
Details
Subject Name
Inc. Country AU
Business Category Private Organization
Serial Number 123 123 124
Country AU
Postal Code 2000
State/Province New South Wales
Locality SYDNEY
Street Address 201 SUSSEX S T
Organization Commonwealth Bank of Australia
Organizational Unit CBA Business System Hosting
Common Name www.commbank.com.au
Issuer Name
Country US
Organization Symantec Corporation
Organizational Unit Symantec Trust Network
Common Name Symantec Class 3 EV SSL CA - G3
Serial Number 1A 9F E9 4B 03 9D E2 9A B6 15 56 69 60 3E 98 AE
Version 3
Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters none
Not Valid Before Monday, 4 May 2015 at 10:00:00 AM Australian Eastern Standard Time
Not Valid After Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time
Public Key Info
Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Parameters none
Public Key 256 bytes : CA B4 74 93 E8 00 22 10 ...
Exponent 65537
Key Size 2048 bits
Key Usage Encrypt, Verify, Wrap, Derive
Signature 256 bytes : 95 32 C3 F0 62 F1 F8 F1 ...
Hide Certificate
OK
Log on
Locate us
Stuff I like
Rates & fees
Latest offers
Find out more


Commonwealth Bank of Australia www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank
Personal
Business
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.
Symantec Corporation has identified www.commbank.com.au as being owned by Commonwealth Bank of Australia in SYDNEY, New South Wales, AU.
VeriSign Class 3 Public Primary Certification Authority - G5
Symantec Class 3 EV SSL CA - G3
www.commbank.com.au
www.commbank.com.au
Expires: Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time
This certificate is valid
Trust
Details
Subject Name
Inc. Country AU
Business Category Private Organization
Serial Number 123 123 124
Country AU
Postal Code 2000
State/Province New South Wales
Locality SYDNEY
Street Address 201 SUSSEX S T
Organization Commonwealth Bank of Australia
Organizational Unit CBA Business System Hosting
Common Name www.commbank.com.au
Issuer Name
Country US
Organization Symantec Corporation
Organizational Unit Symantec Trust Network
Common Name Symantec Class 3 EV SSL CA - G3
Serial Number 1A 9F E9 4B 03 9D E2 9A B6 15 56 69 60 3E 98 AE
Version 3
Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters none
Not Valid Before Monday, 4 May 2015 at 10:00:00 AM Australian Eastern Standard Time
Not Valid After Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time
Public Key Info
Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Parameters none
Public Key 256 bytes : CA B4 74 93 E8 00 22 10 ...
Exponent 65537
Key Size 2048 bits
Key Usage Encrypt, Verify, Wrap, Derive
Signature 256 bytes : 95 32 C3 F0 62 F1 F8 F1 ...
How did my browser know that this is a valid cert?
Hide Certificate
OK
Log on
Locate us
Stuff I like
Rates & fees
Latest offers
Find out more >

Domain Name Certification

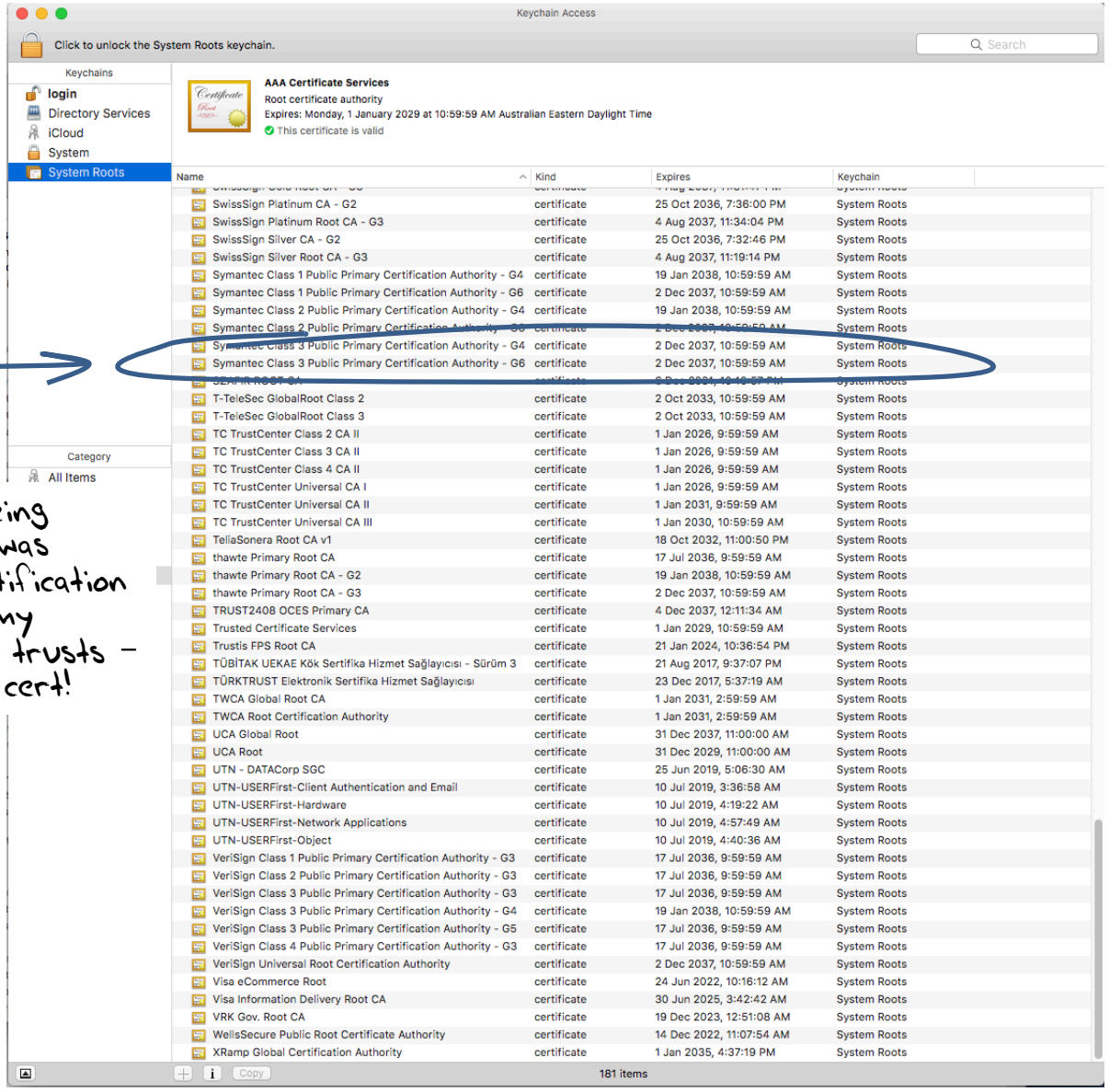
- The Commonwealth Bank of Australia has generated a key pair
- And they passed a certificate signing request to a company called “Symantec”
- Who is willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value
- So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to www.commbank.com.au, as long as I am prepared to trust Symantec and the certificates that they issue

Domain Name Certification

- The Commonwealth Bank of Australia has generated a key pair
- And they passed a certificate signing request to a company called “Symantec”
- Who is willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value
- So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to www.commbank.com.au, as long as I am prepared to trust Symantec and the certificates that they issue

Why should i trust them?

Local Trust

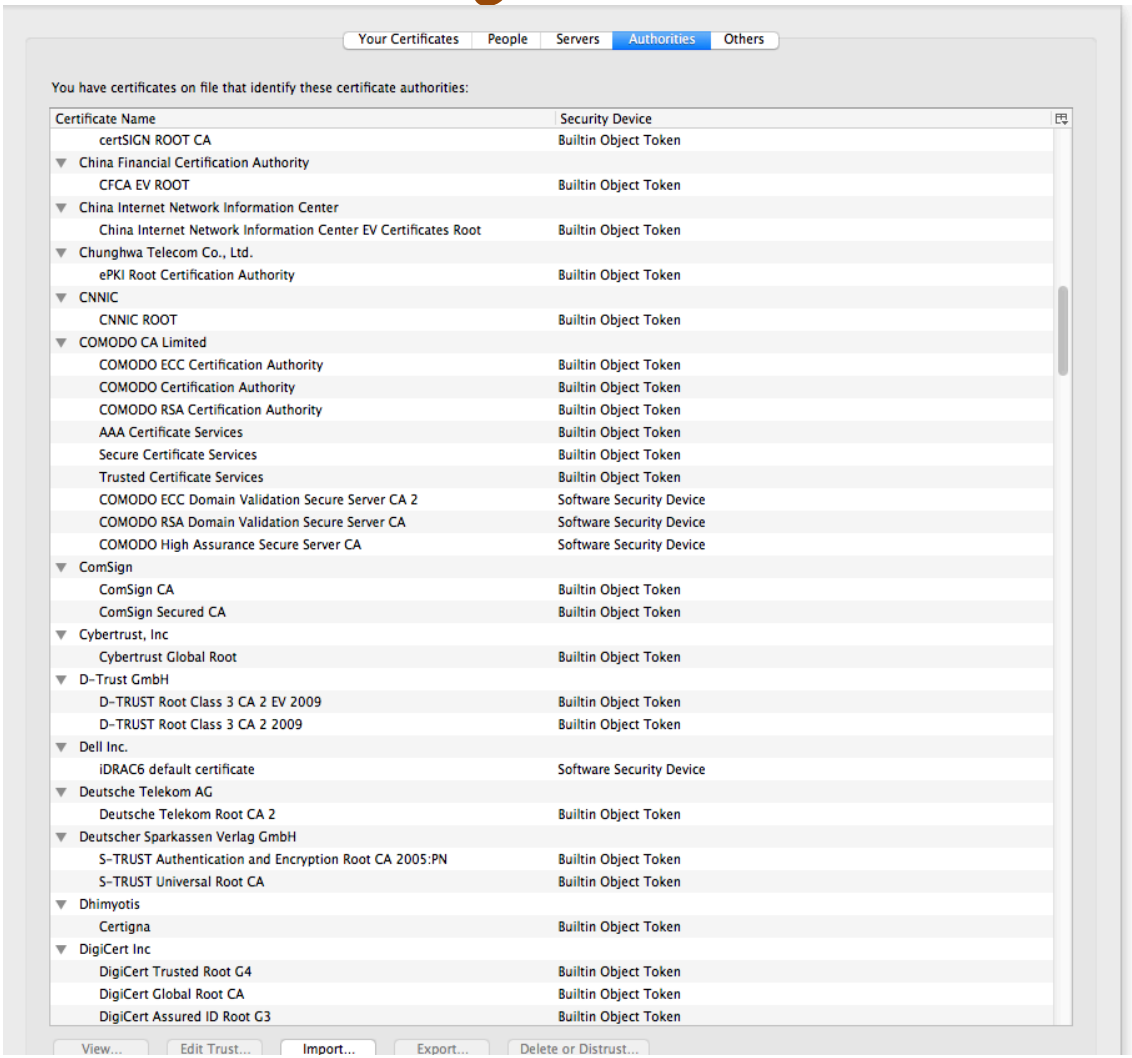


The cert I'm being asked to trust was issued by a certification authority that my browser already trusts - so I trust that cert!

Local Trust or Local Credulity*?

That's a big list of people to Trust

Are they all trustable?



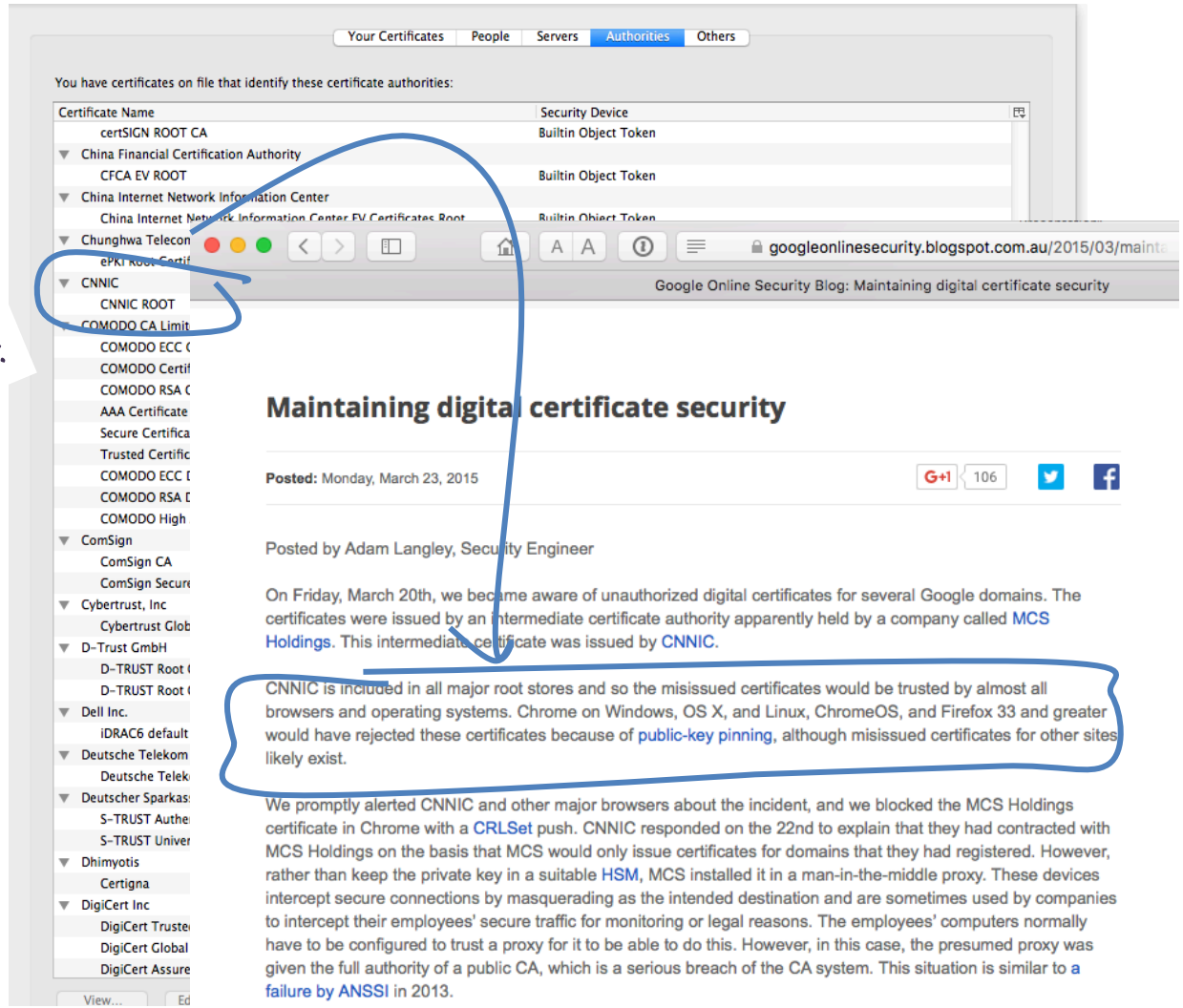
* cre·du·li·ty
/krəˈd(y)o͞olədē/
noun
a tendency to be too ready to believe that something is real or true.

Local Credulity

That's a big list of people to Trust

Are they all trustable?

Evidently Not!

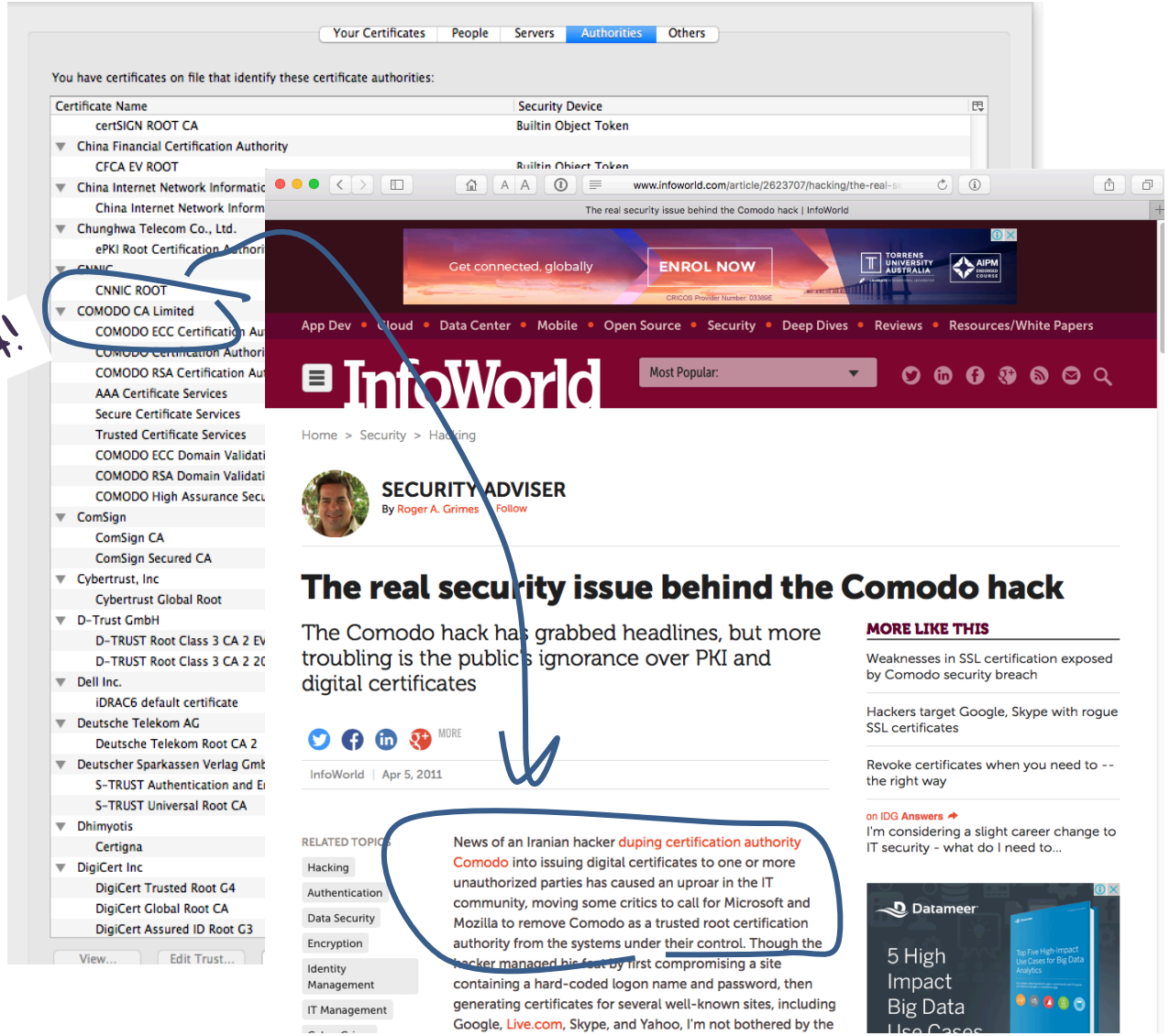


Local Credulity

That's a big list of people to Trust

Are they all trustable?

Evidently Not!



But my bank used Symantec

as their Certificate Authority

And Symantec NEVER lie in the certificates they issue

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:

With unpleasant consequences
when it all goes wrong

With unpleasant consequences when it all goes wrong

talk
ow

Cuba aimed at U.S.
her 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

sters helped ignited
untry's 45-member

ing 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 not resp

In th
is most
reckon
HACKER,

net attacks, this
ment of

International Herald Tribune
Sep 13, 2011 Front Page

What's going wrong here?

- The TLS handshake cannot specify WHICH CA should be used to validate the digital certificate
- 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 to validate the digital certificate
- 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 should be used to validate a digital certificate
- You

WOW! That's awesomely bad!

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!

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 you 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

Than what 'wins' in the market?

?

In a commercial environment

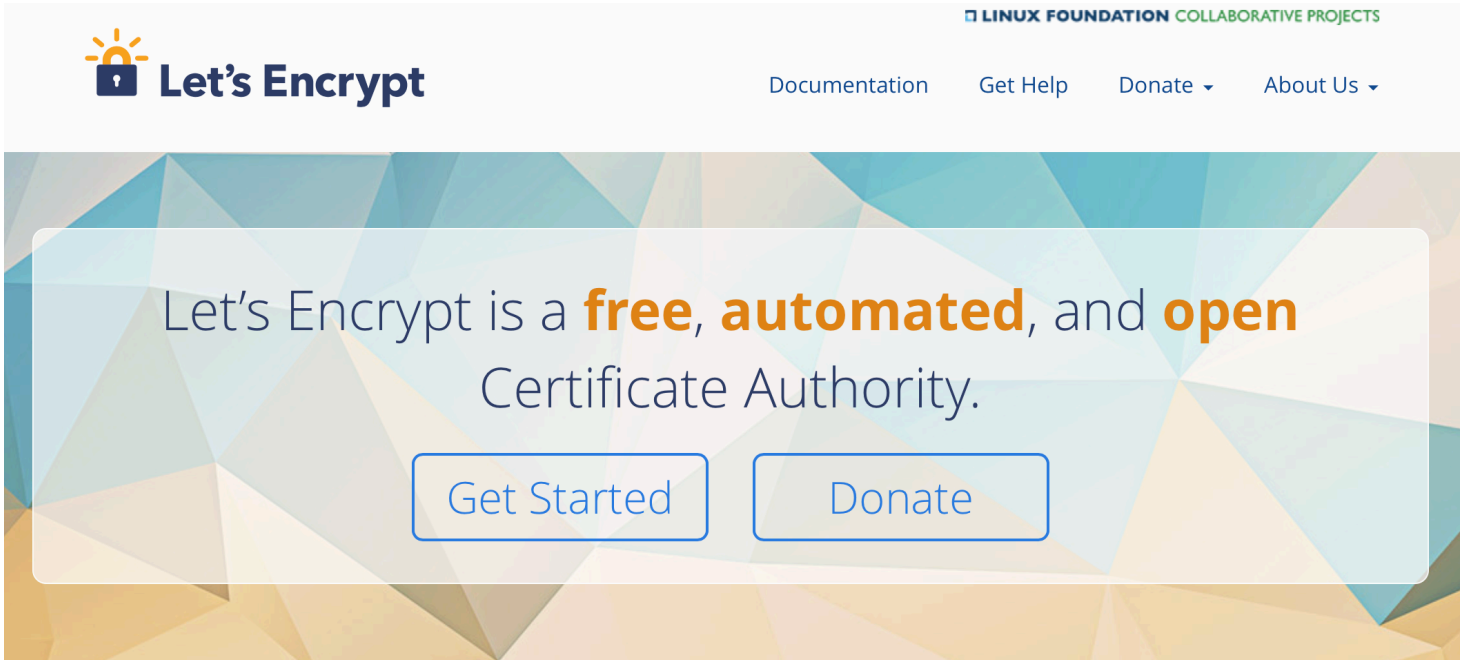
Where CA's compete with each other for market share

And quality offers no protection

Than what 'wins' in the market?

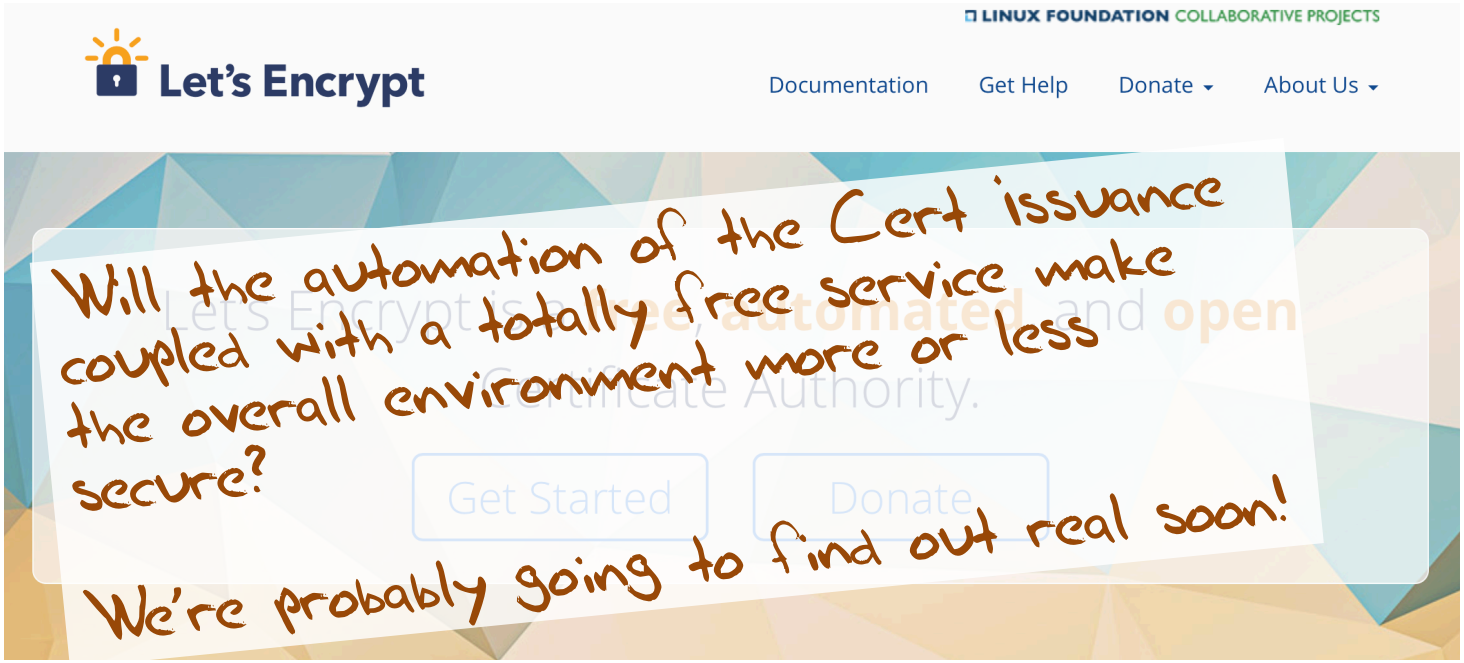
Where now?

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



Where now?

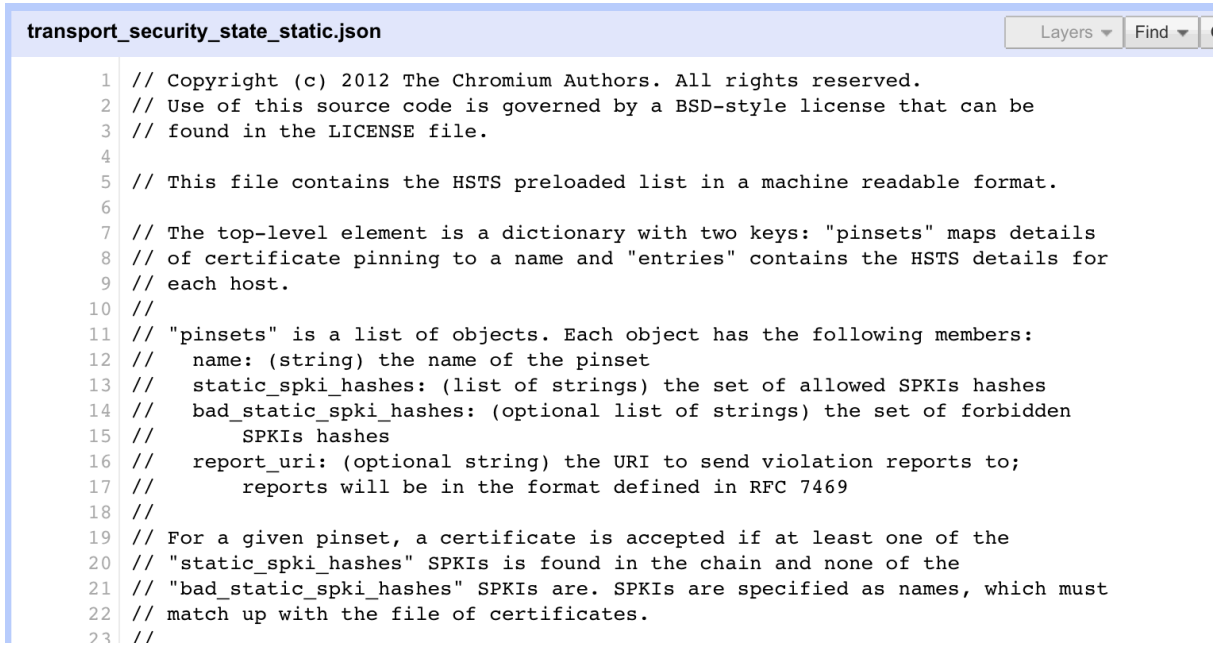
Option A: Take all the money out of the system!



Where now?

Option B: White Listing and Pinning with HSTS

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



Where now?

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!

Layers Find

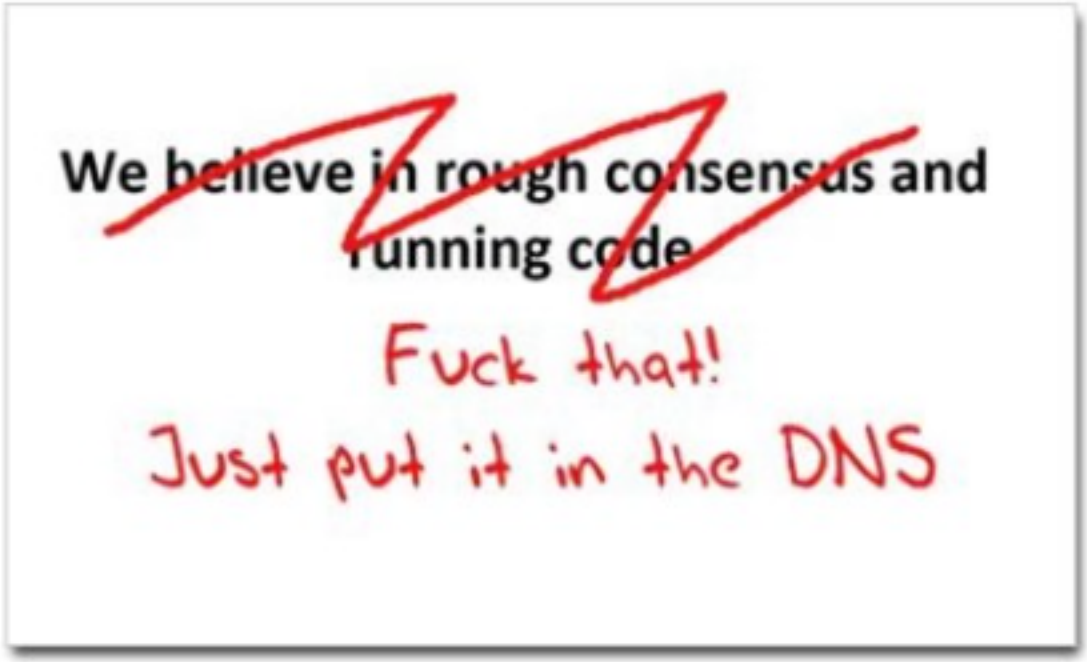
```
// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// 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.
//
```

Where now?

Option C: Use the DNS!



www.cafepress.com/nxdomain

Seriously

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

Seriously

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 (pinning record)?

Seriously

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?

Seriously

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?

Seriously

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

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?

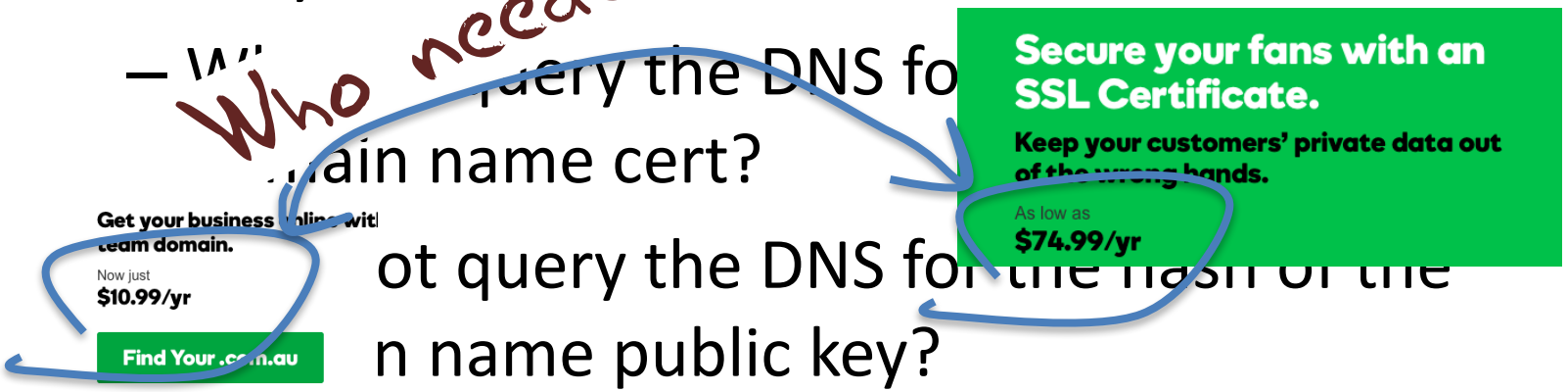
Who needs CA's anyway?

Seriously

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

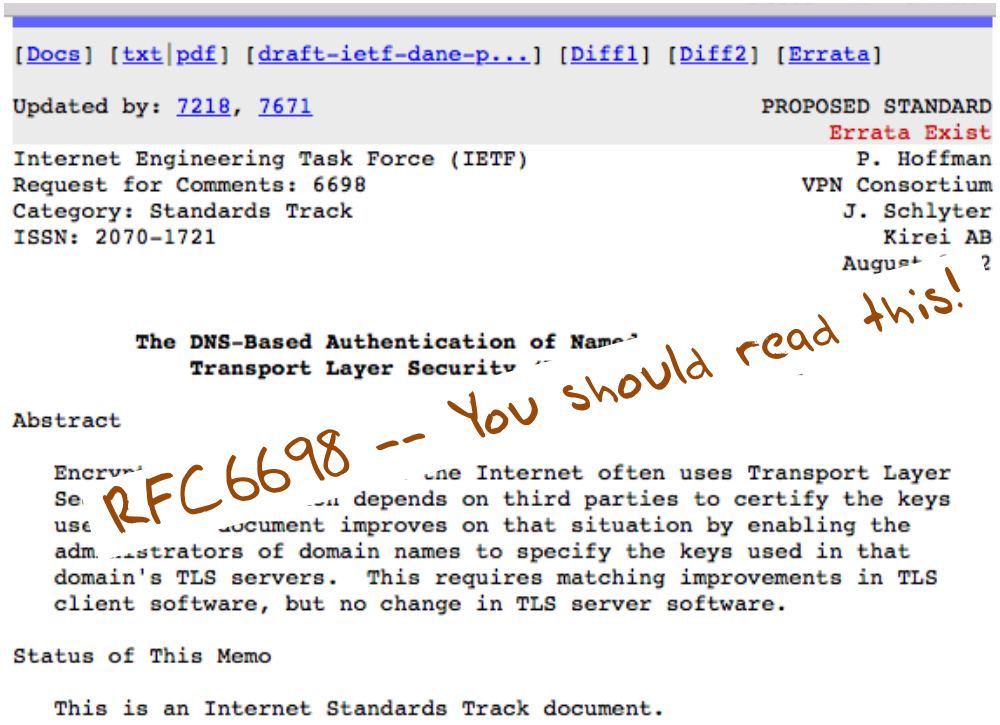
- Why not query the DNS for the HSTS record?
- Why not query the DNS for the issuer CA?
- Why not query the DNS for the domain name cert?
- Why not query the DNS for the hash of the domain name public key?

Who needs CA's anyway?



DANE

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



[Docs] [txt|pdf] [draft-ietf-dane-p...] [Diff1] [Diff2] [Errata]

Updated by: 7218, 7671 — PROPOSED STANDARD
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
August ?

The DNS-Based Authentication of Name[illegible]
Transport Layer Security[illegible]

RFC6698 -- You should read this!

Abstract

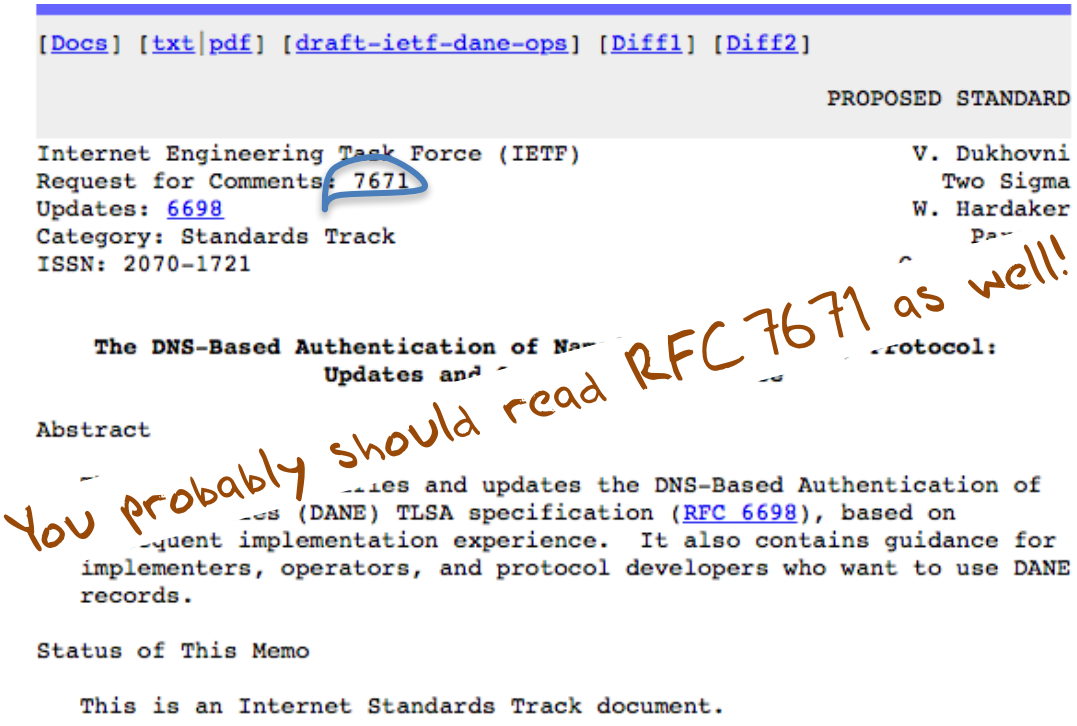
Encry[illegible]ne Internet often uses Transport Layer Se[illegible] depends on third parties to certify the keys use[illegible]ocument improves on that situation by enabling the adm[illegible]istrators of domain names to specify the keys used in that domain's TLS servers. This requires matching improvements in TLS client software, but no change in TLS server software.

Status of This Memo

This is an Internet Standards Track document.

DANE

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



[Docs] [txt|pdf] [draft-ietf-dane-ops] [Diff1] [Diff2]

PROPOSED STANDARD

Internet Engineering Task Force (IETF)
Request for Comments: 7671
Updates: 6698
Category: Standards Track
ISSN: 2070-1721

V. Dukhovni
Two Sigma
W. Hardaker
Pa-

The DNS-Based Authentication of Na- -rotocol: Updates and -

Abstract

-ies and updates the DNS-Based Authentication of -s (DANE) TLSA specification (RFC 6698), based on -quent implementation experience. It also contains guidance for implementers, operators, and protocol developers who want to use DANE records.

Status of This Memo

This is an Internet Standards Track document.

You probably should read RFC 7671 as well!

DANE

TLSA RR

2.3. TLSA RR Examples

An example of a hashed (SHA-256) association of a PKIX CA certificate:

```
_443._tcp.www.example.com. IN TLSA (
    0 0 1 d2abde240d7cd3ee6b4b28c54df034b9
          7983a1d16e8a410e4561cb106618e971 )
```

CA Cert Hash

An example of a hashed (SHA-512) subject public key association of a PKIX end entity certificate:

```
_443._tcp.www.example.com. IN TLSA
    1 1 2 92003ba34942dc74152e2f2c408d29ec
          a5a520e7f2e06bb944f4dca346baf63c
          1b177615d466f6c4b71c216a50292bd5
          8c9ebdd2f74e38fe51ffd48c43326cbc )
```

EE Cert Hash

An example of a full certificate association of a PKIX trust anchor:

```
_443._tcp.www.example.com. IN TLSA
    2 0 0 30820307308201efa003020102020... )
```

Trust Anchor

EECert TLSA record generation

```
; Convert the public key certificate to DER format
; Generate the SHA256 hash
; Add DNS gunk!

$ /usr/bin/openssl x509 -in /usr/local/etc/letsencrypt/live/www.dotnxdomain.net/cert.pem -outform DER |
/usr/bin/openssl sha256 |
cut -d ' ' -f 2 |
awk '{print "_443._tcp.www.dotnxdomain.net  IN TLSA 3 0 1 " $1}'

_443._tcp.www.dotnxdomain.net. 899 IN     TLSA  3 0 1 D42101BCCE941D22E8E467C5D75E77EC4A7B8B7C9366C6A878CB4E15 7E602F17
```

```
$ dig +dnssec TLSA _443._tcp.www.dotnxdomain.net.

_443._tcp.www.dotnxdomain.net. 899 IN     TLSA  3 0 1 D42101BCCE941D22E8E467C5D75E77EC4A7B8B7C9366C6A878CB4E15 7E602F17
_443._tcp.www.dotnxdomain.net. 899 IN     RRSIG TLSA 13 5 900 20200724235900 20170122043100 56797 www.dotnxdomain.net.
dUYD1sMIpBc6RsUhturFzz5G8qX6oaDGRzaD/q6n+YJi2kqzDfWZls6F 3X1mXdpeQQYz52yOUOcdWvFRO9TQZQ==
```

SPKI TLSA record generation

```
; Generate the public key
; Convert it to DER format
; Generate the SHA256 hash
; Add DNS gunk!
```

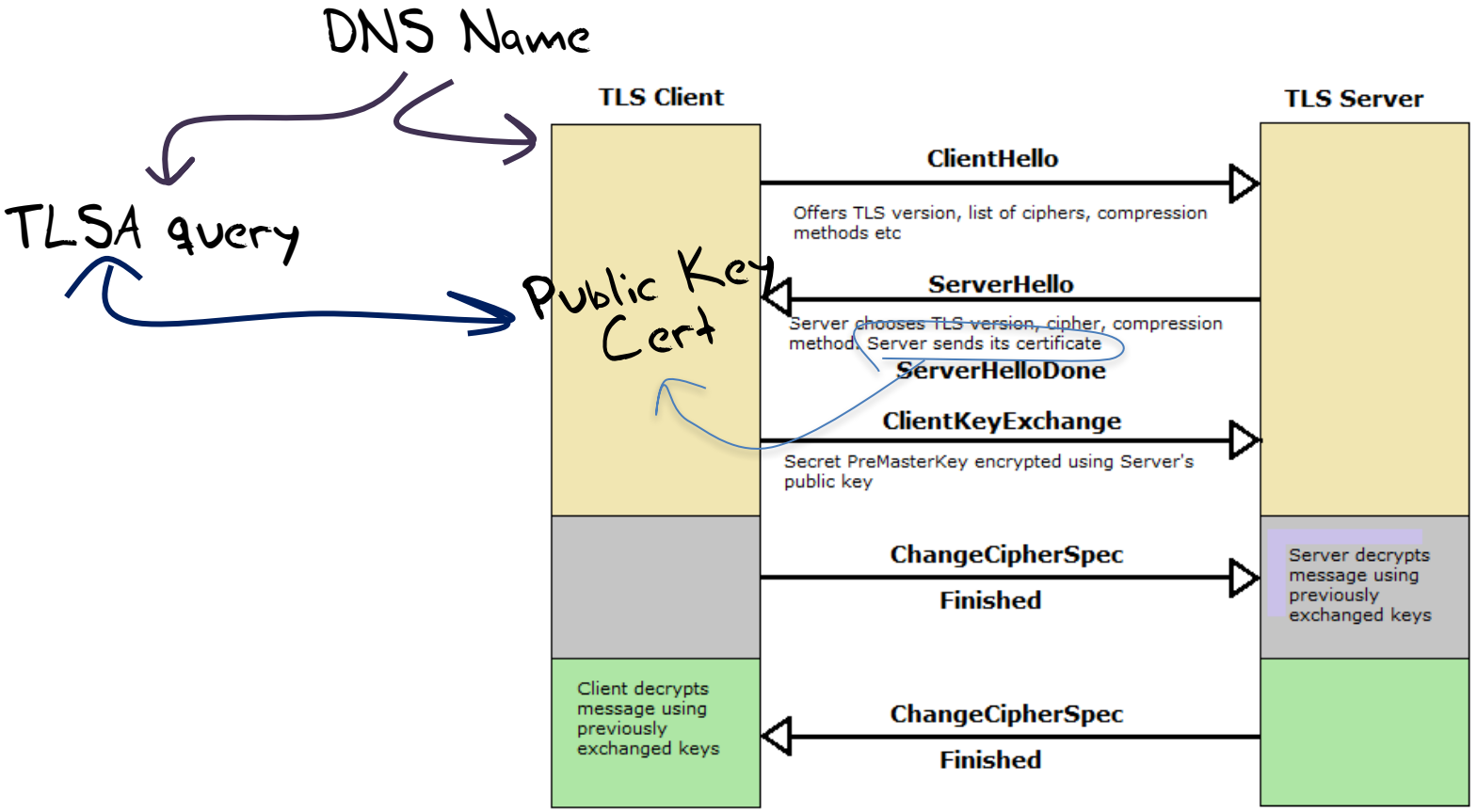
```
$ /usr/bin/openssl x509 -in /usr/local/etc/letsencrypt/live/www.dotnxdomain.net/cert.pem -pubkey -noout |
openssl rsa -pubin -outform der |
/usr/bin/openssl sha256 |
cut -d ' ' -f 2 |
awk '{ print "_443._tcp.www.ndotnxdomain.net IN TLSA 3 1 1 " $1}'

_443._tcp.www.ndotnxdomain.net IN TLSA 3 1 1 df3a810d998cfddf8fa935ed33065ee27a67747366e2da40ddefef2b3a2032eb
```

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 its called DNSSEC!

DNSSEC Interlocking Signatures

. (root)

. Key-Signing Key – signs over

. Zone-Signing Key – signs over

DS for .com (Key-Signing Key)

.com

.com Key-Signing Key – signs over

.com Zone-Signing Key – signs over

DS for example .com (Key-Signing Key)

.example.com

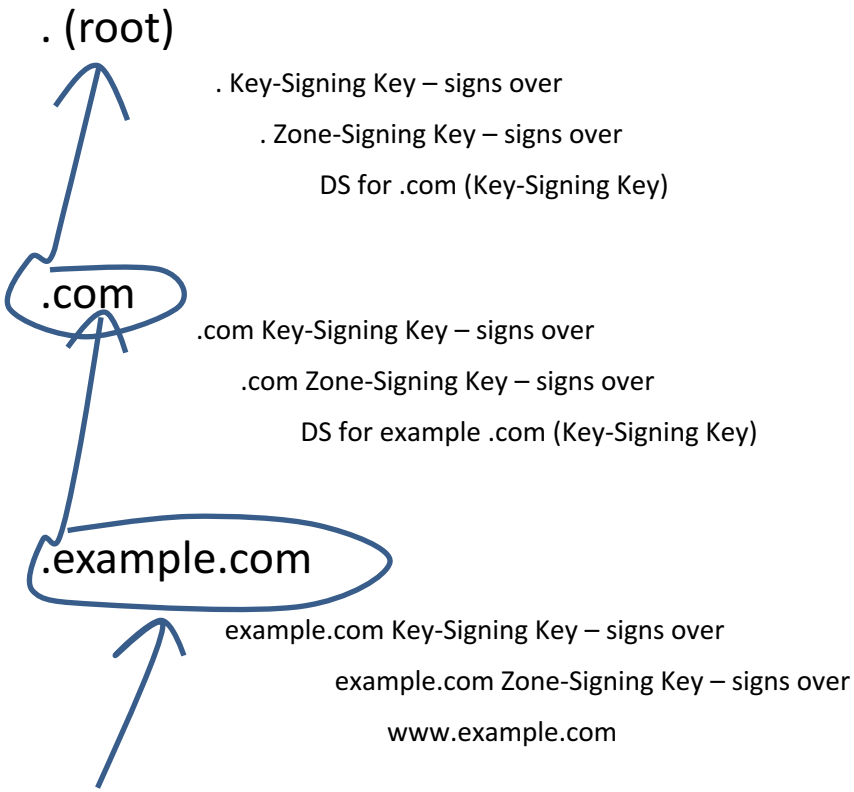
example.com Key-Signing Key – signs over

example.com Zone-Signing Key – signs over

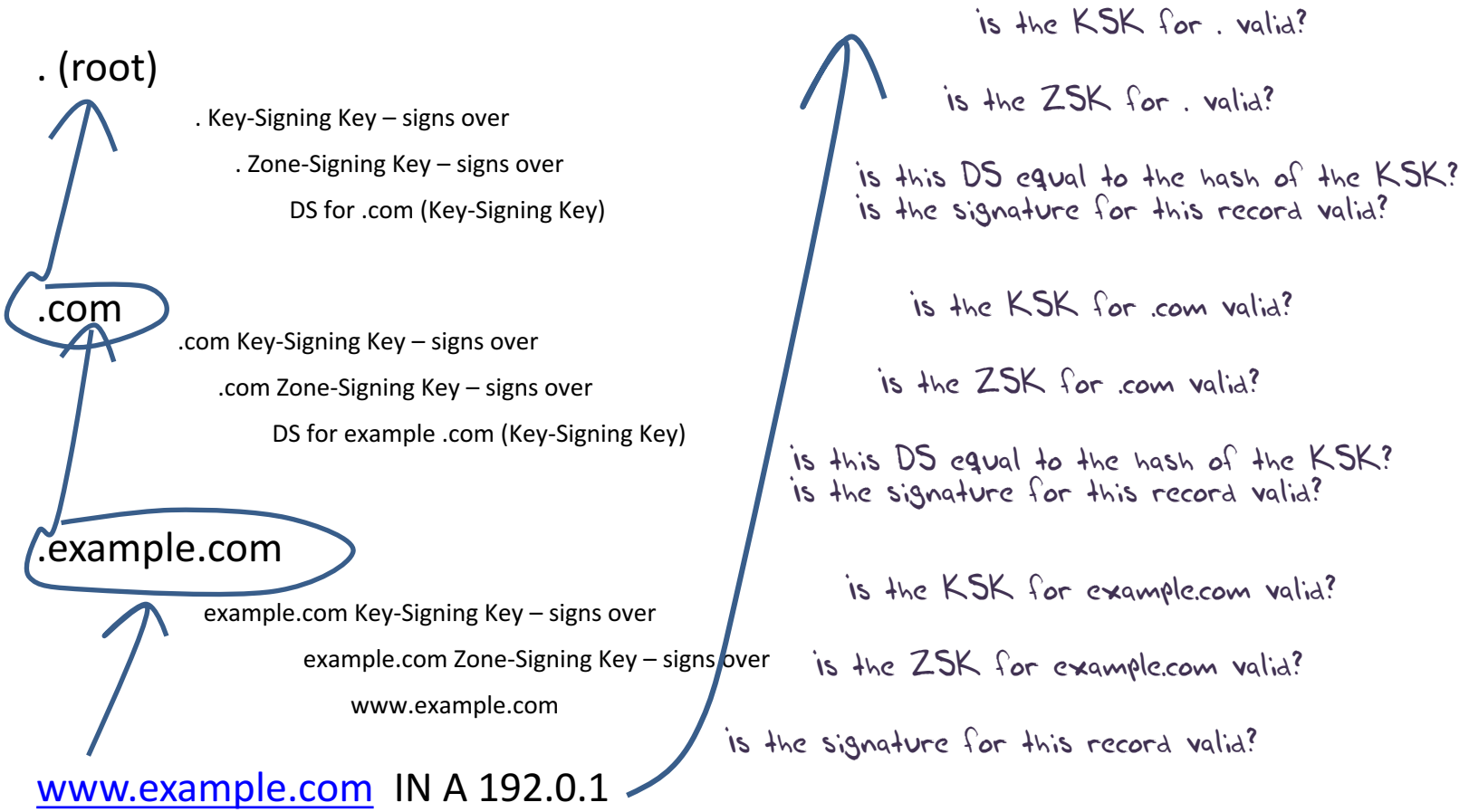
www.example.com

www.example.com

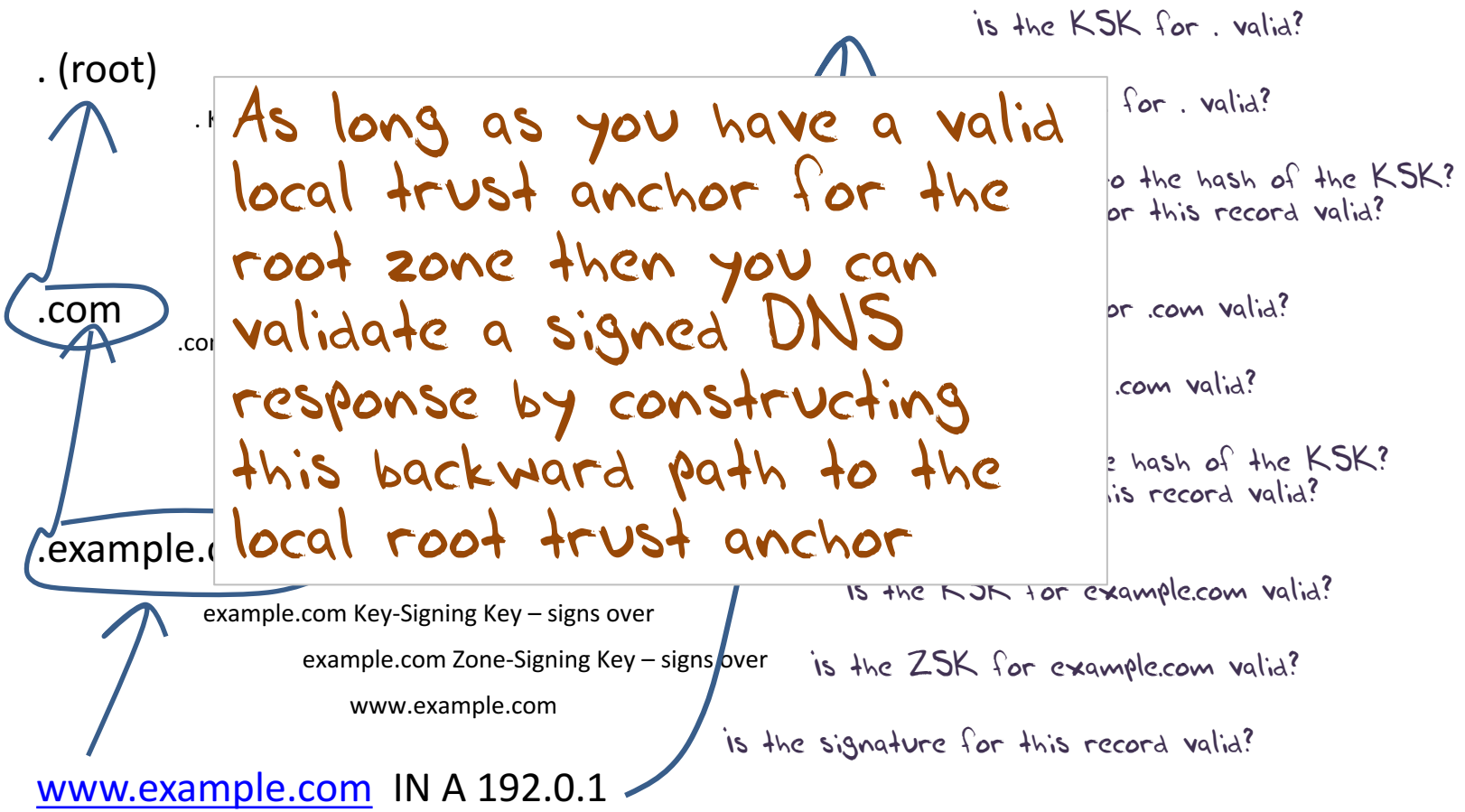
DNSSEC Interlocking Signatures



DNSSEC Interlocking Signatures



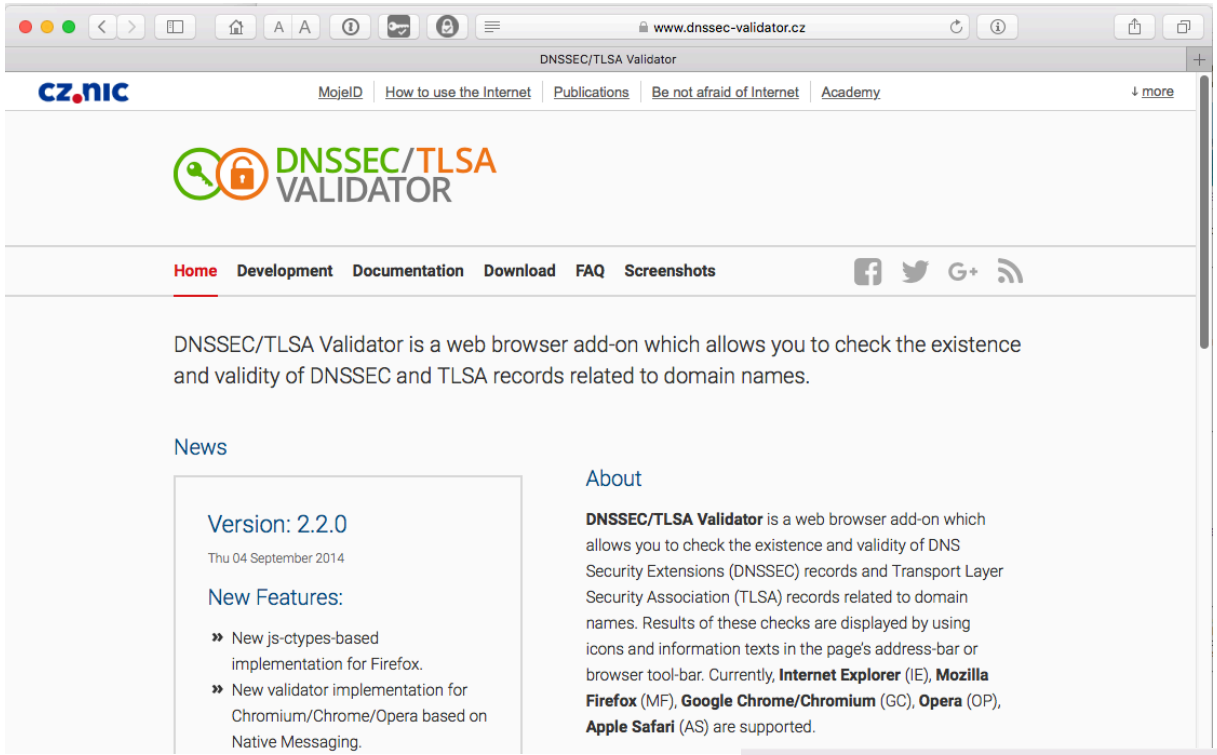
DNSSEC Interlocking Signatures

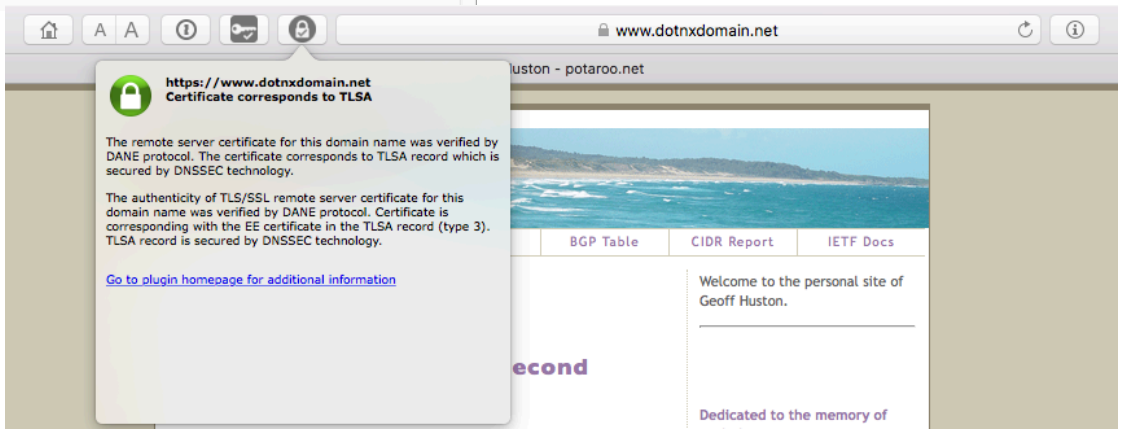


DANE + DNSSEC

- 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 Does DNS via a Browser Extension



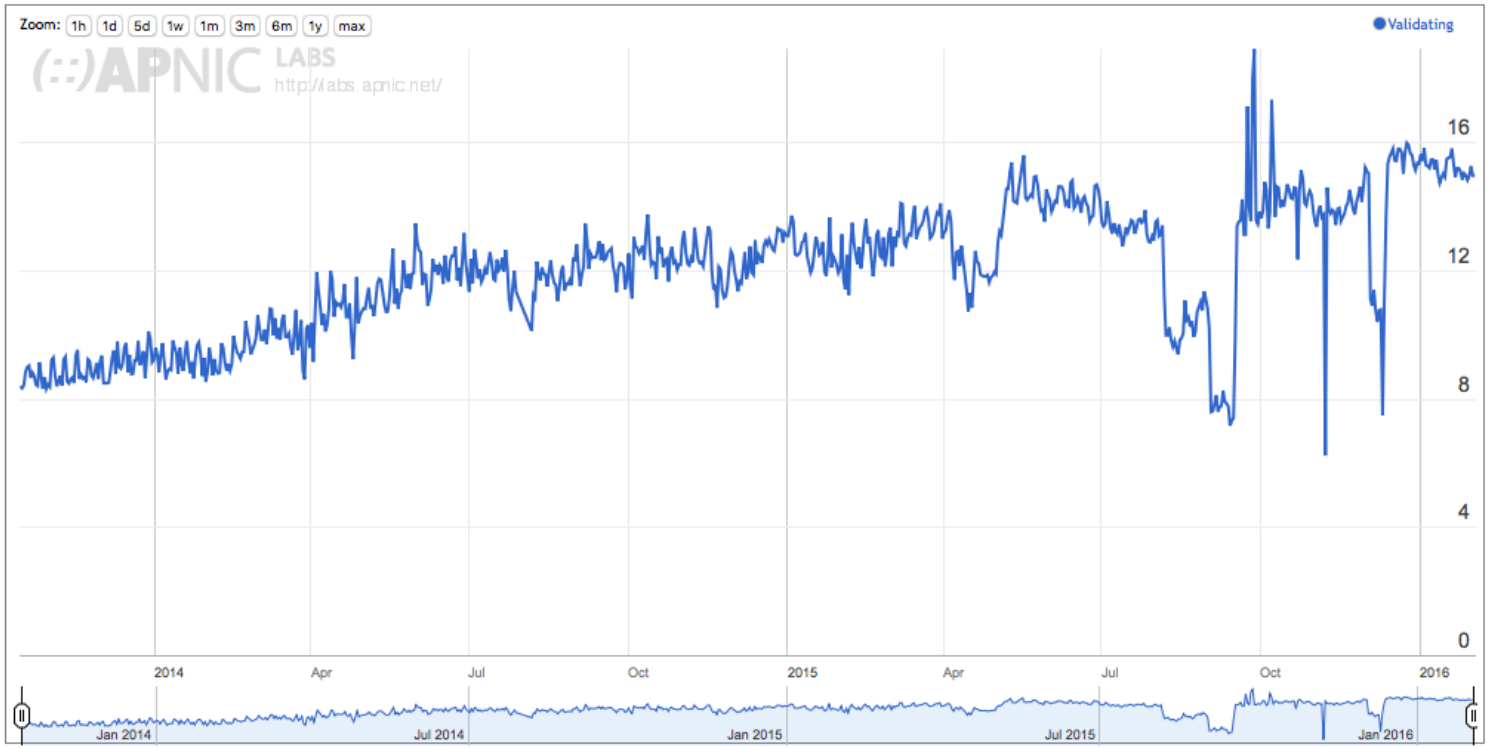


So we need DNSSEC as well as DANE…

How much DNSSEC Validation is out there?

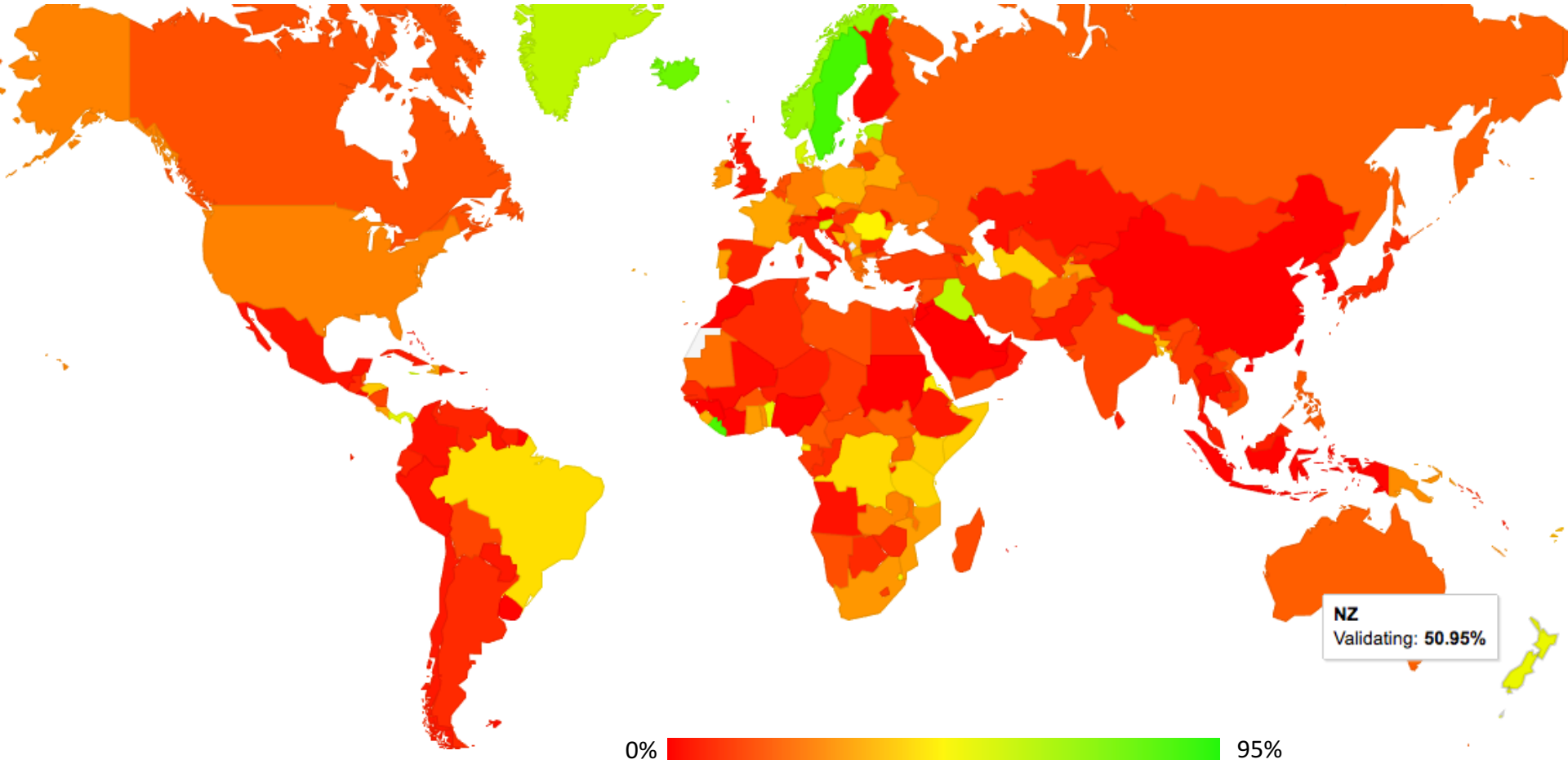
Do we do DNSSEC Validation?

Use of DNSSEC Validation for World (XA)



stats.labs.apnic.net/dnssec/XA

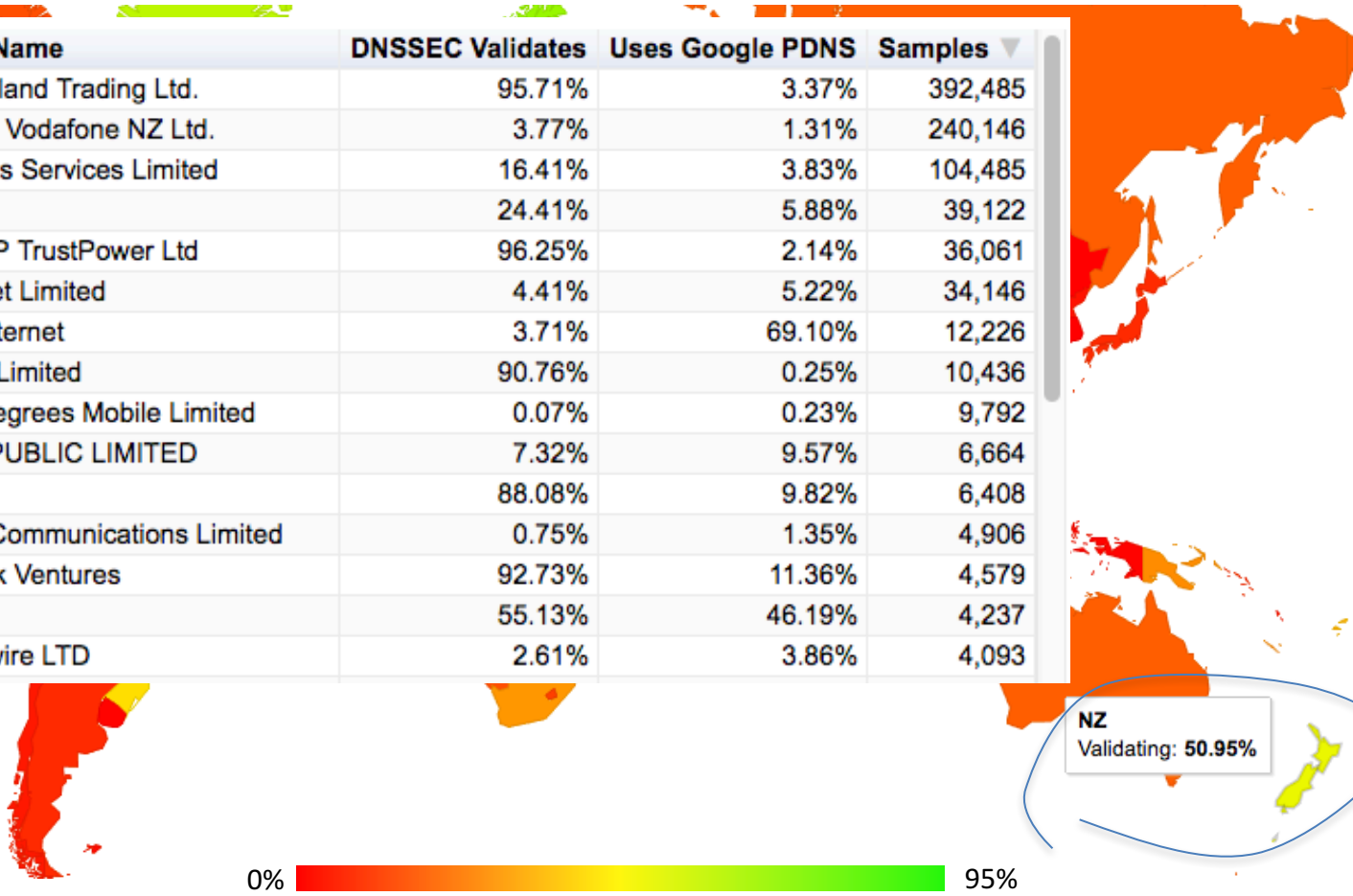
Do we do DNSSEC Validation?



stats.labs.apnic.net/dnssec/XA

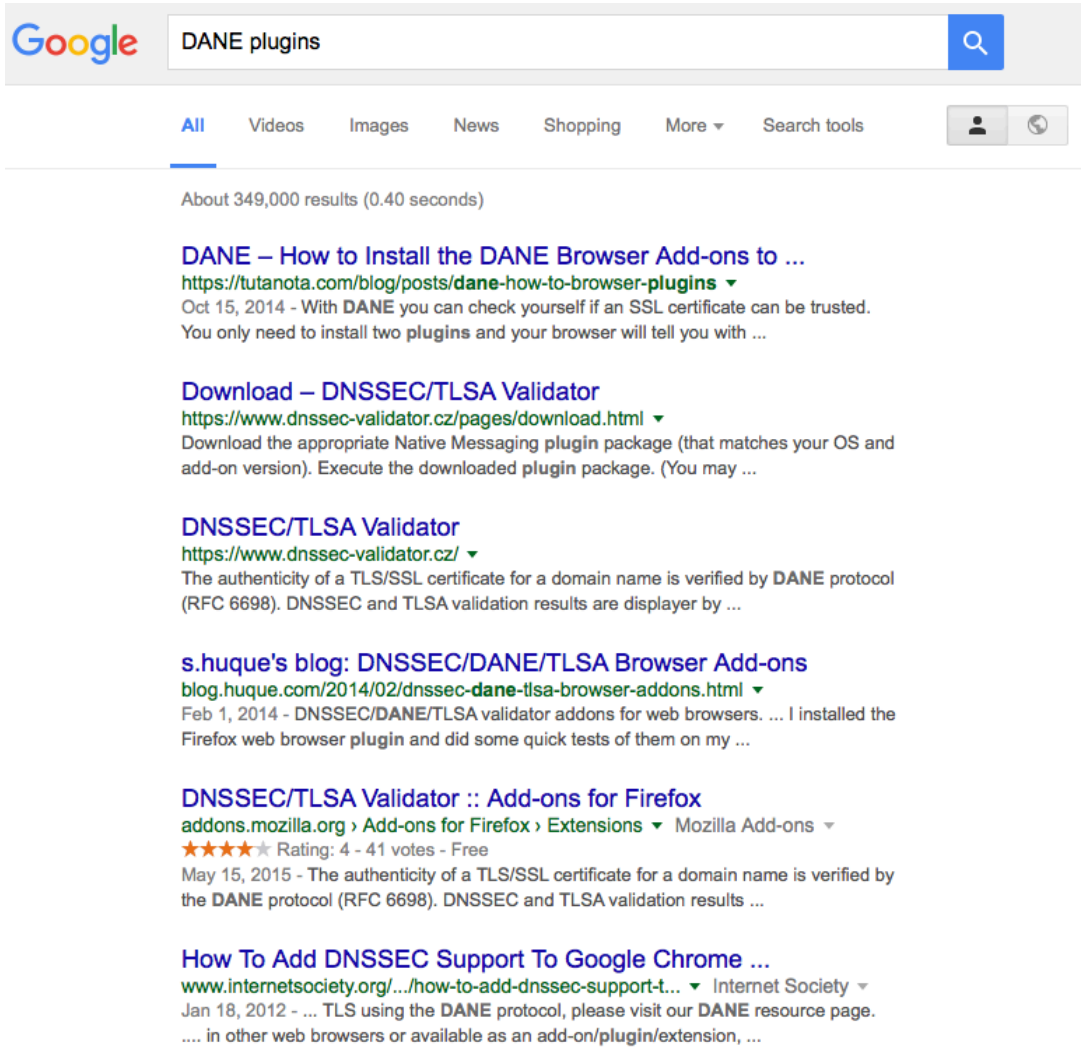
Do we do DNSSEC Validation?

ASN	AS Name	DNSSEC Validates	Uses Google PDNS	Samples
AS4771	SPARKNZ Spark New Zealand Trading Ltd.	95.71%	3.37%	392,485
AS9500	VODAFONE-TRANSIT-AS Vodafone NZ Ltd.	3.77%	1.31%	240,146
AS9790	CALLPLUS-NZ-AP CallPlus Services Limited	16.41%	3.83%	104,485
AS4768	CLIX-NZ TelstraClear Ltd	24.41%	5.88%	39,122
AS55850	TRUSTPOWERLTD-AS-AP TrustPower Ltd	96.25%	2.14%	36,061
AS23655	SNAP-NZ-AS Snap Internet Limited	4.41%	5.22%	34,146
AS4648	NZIX-2 Global-Gateway Internet	3.71%	69.10%	12,226
AS58600	FLIP-AS-AP Flip Services Limited	90.76%	0.25%	10,436
AS38793	NZCOMMS-AS-AP Two Degrees Mobile Limited	0.07%	0.23%	9,792
AS133579	MYREPNZ-AS-AP MYREPUBLIC LIMITED	7.32%	9.57%	6,664
AS9876	AIRNET-HB-AS-AP NOW	88.08%	9.82%	6,408
AS55872	BAYCITY-AS-AP BayCity Communications Limited	0.75%	1.35%	4,906
AS133124	SPARKVENT-AS-AP Spark Ventures	92.73%	11.36%	4,579
AS55853	MEGATEL-AS-AP Megatel	55.13%	46.19%	4,237
AS45267	LIGHTWIRE-AS-AP Lightwire LTD	2.61%	3.86%	4,093



stats.labs.apnic.net/dnssec/XA

Where now?



Browser vendors appear to be dragging the chain on DANE support

DANE exists today as plug-ins rather than a core functionality

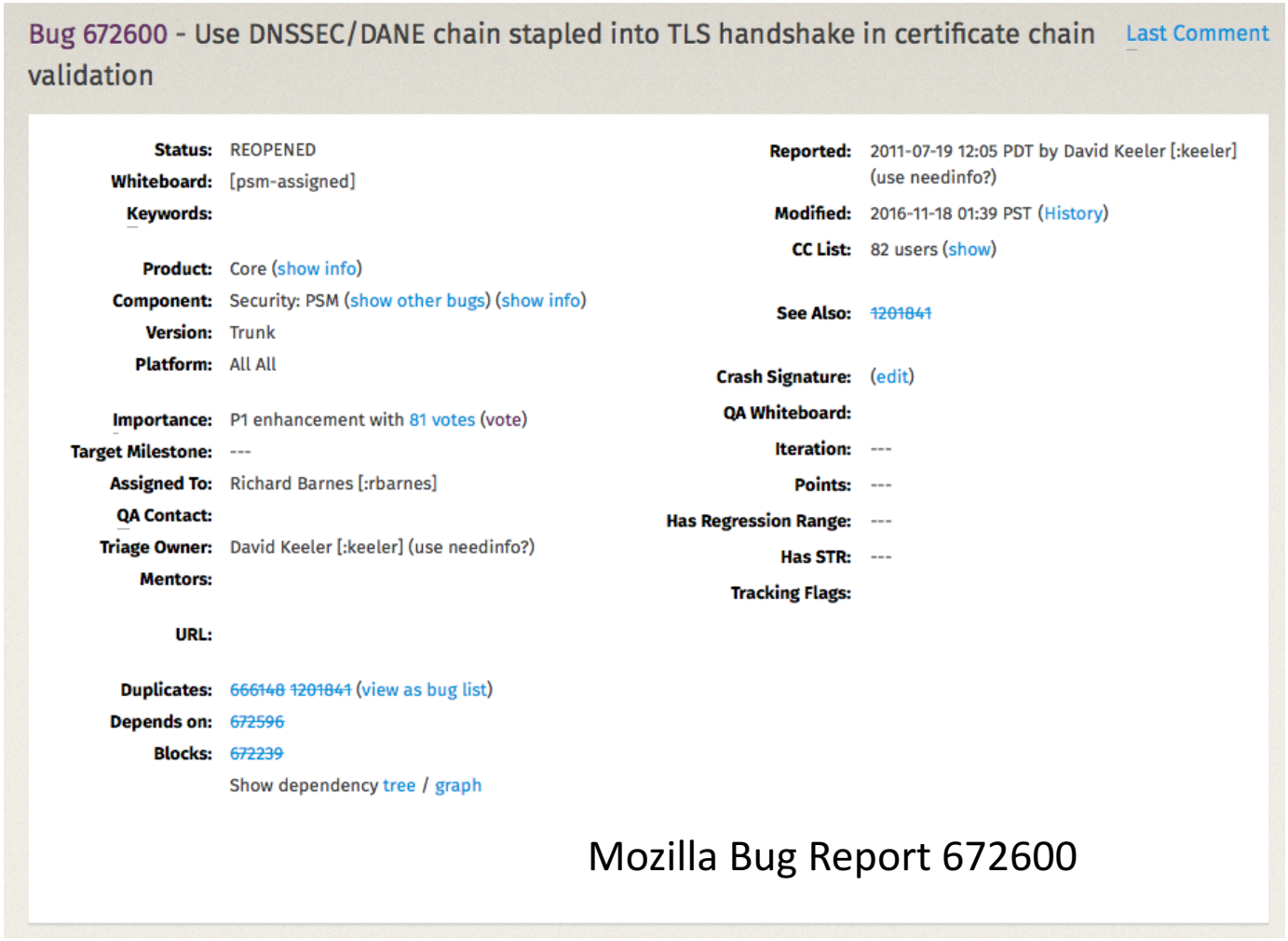
Cynically, one could observe that fast but insecure is the browser vendors' current preference!

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

Faster DANE with Stapling



Bug 672600 - Use DNSSEC/DANE chain stapled into TLS handshake in certificate chain validation Last Comment

Status:	REOPENED	**Reported:**	2011-07-19 12:05 PDT by David Keeler [:keeler] (use needinfo?)
Whiteboard:	[psm-assigned]	**Modified:**	2016-11-18 01:39 PST (History)
Keywords:		**CC List:**	82 users (show)
Product:	Core (show info)		
Component:	Security: PSM (show other bugs) (show info)	**See Also:**	~~1201841~~
Version:	Trunk		
Platform:	All All	**Crash Signature:**	(edit)
		QA Whiteboard:	
Importance:	P1 enhancement with 81 votes (vote)	**Iteration:**	---
Target Milestone:	---	**Points:**	---
Assigned To:	Richard Barnes [:rbarnes]	**Has Regression Range:**	---
QA Contact:		**Has STR:**	---
Triage Owner:	David Keeler [:keeler] (use needinfo?)	**Tracking Flags:**	
Mentors:			
URL:			
Duplicates:	~~666148~~ ~~1201841~~ (view as bug list)		
Depends on:	~~672596~~		
Blocks:	~~672239~~		
	Show dependency tree / graph		

Mozilla Bug Report 672600

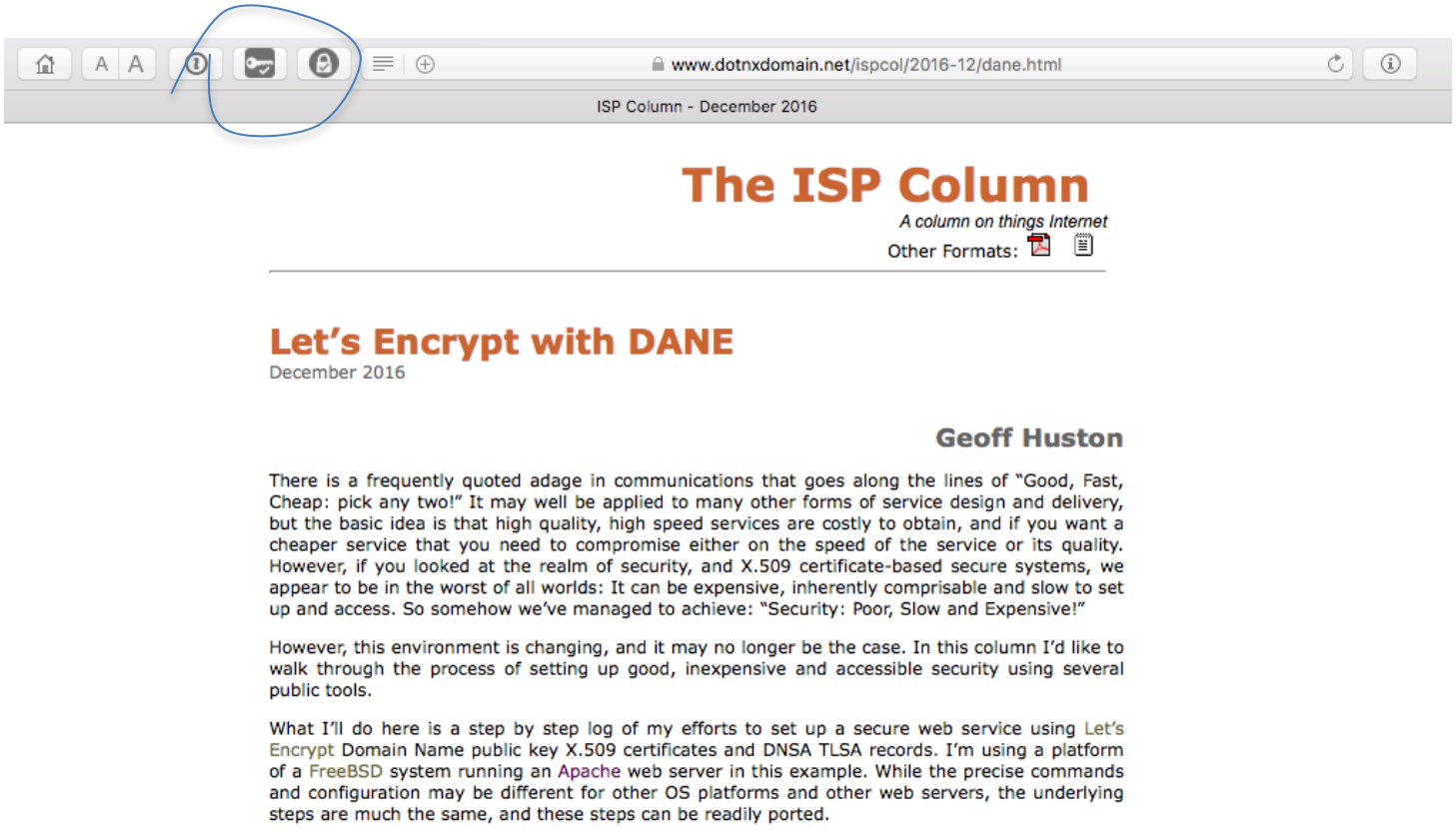
Where now?

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

What this can offer is robust, affordable, accessible security without the current overheads of high priced vanity CA offerings

Let's Do it!



What Let's Encrypt and DNSSEC offers is robust, affordable, accessible security without the current overheads of high priced vanity CA offerings

That's it!

Questions?