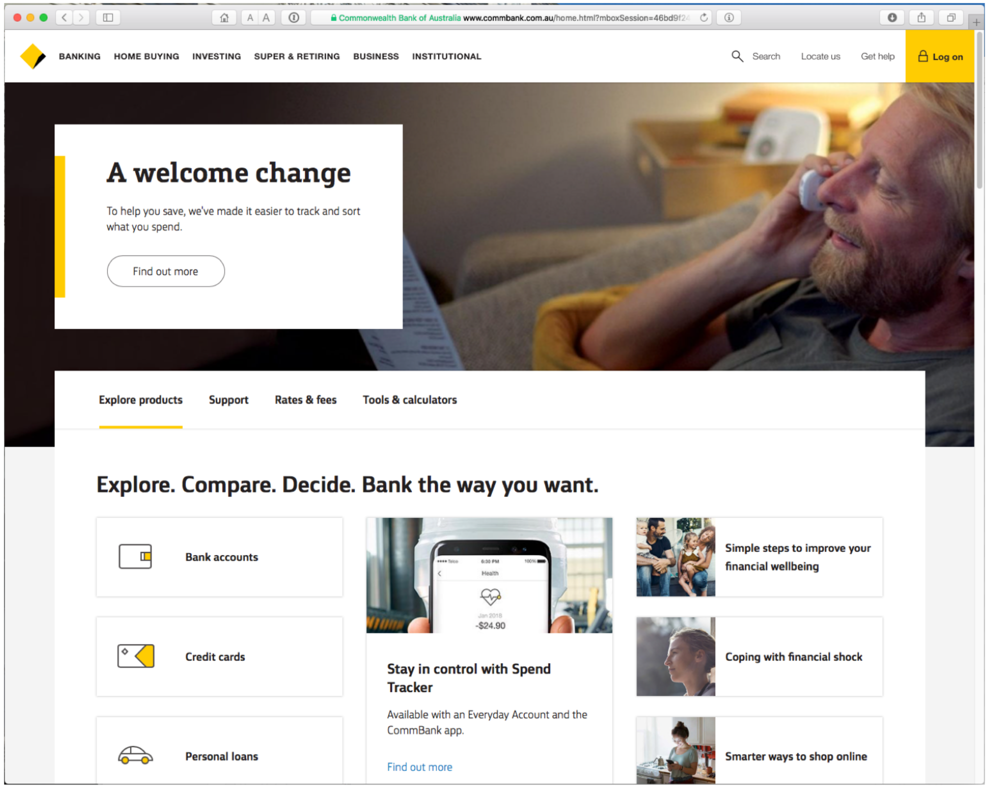
Why “Dane”?

Geoff Huston
Chief Scientist, APNIC

Which Bank?



A welcome change

To help you save, we've made it easier to track and sort what you spend.

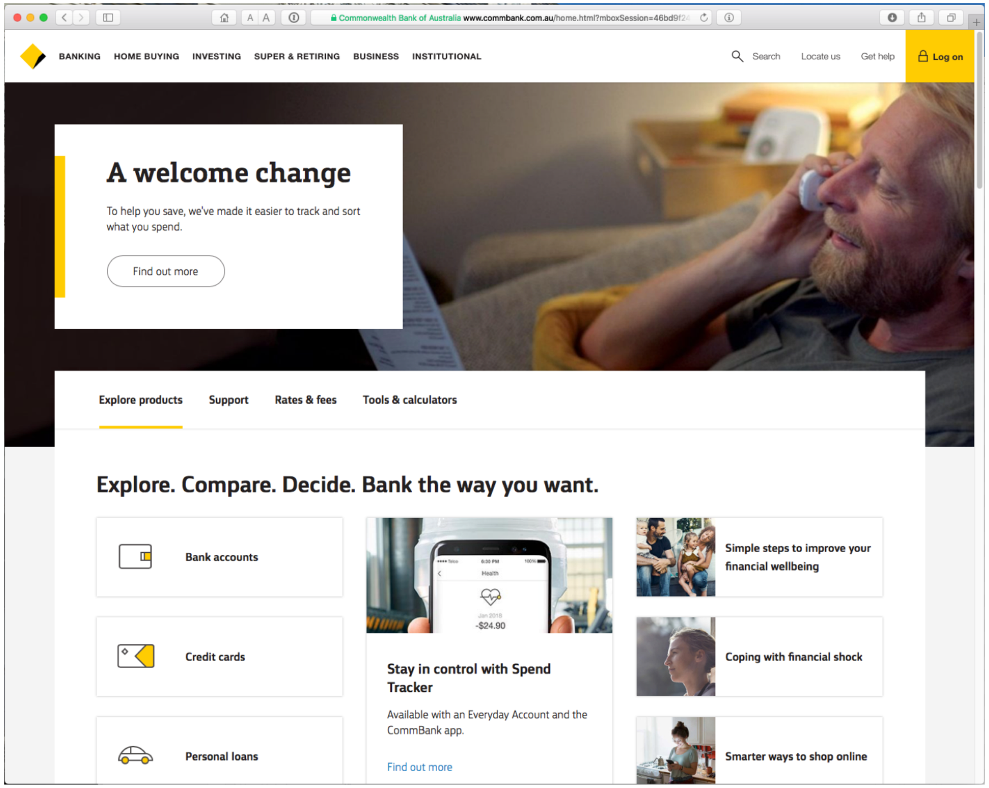
Explore. Compare. Decide. Bank the way you want.

Stay in control with Spend Tracker

Available with an Everyday Account and the CommBank app.

Find out more
Simple steps to improve your financial wellbeing
Coping with financial shock
Smarter ways to shop online

Which Bank? My Bank!



A welcome change

To help you save, we've made it easier to track and sort what you spend.

Find out more

Explore. Compare. Decide. Bank the way you want.

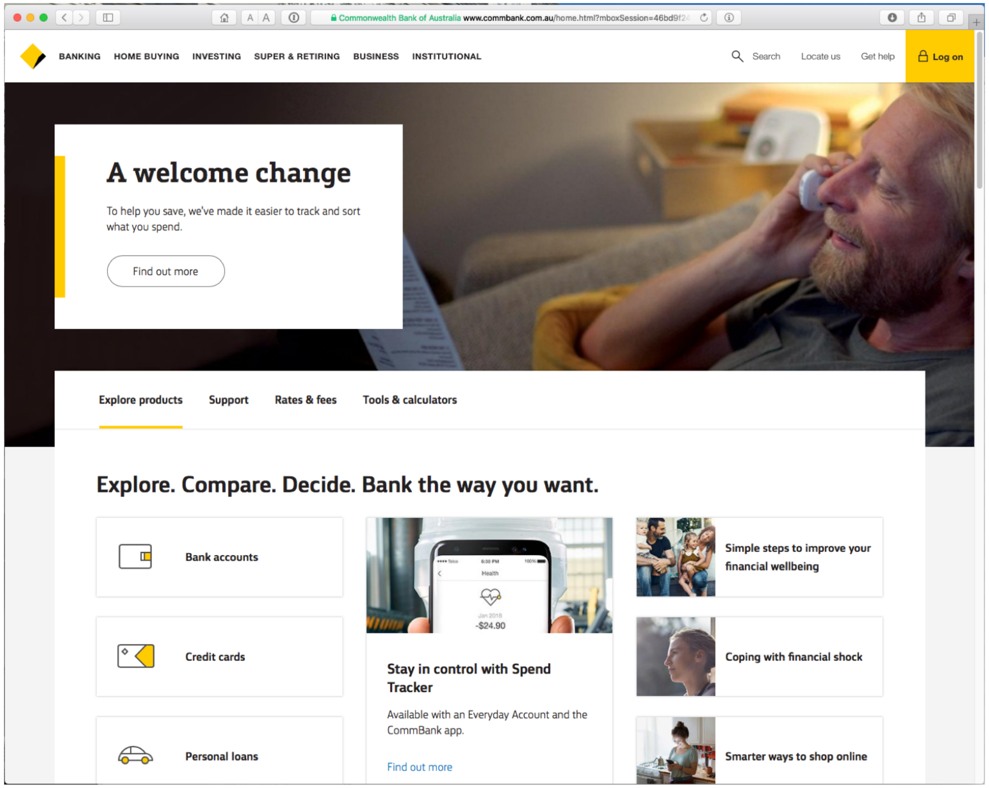
Stay in control with Spend Tracker

Available with an Everyday Account and the CommBank app.

Find out more

Health
-$24.90
Simple steps to improve your financial wellbeing
Coping with financial shock
Smarter ways to shop online

Which Bank? My Bank!



A welcome change

To help you save, we've made it easier to track and sort what you spend.

Explore. Compare. Decide. Bank the way you want.

Stay in control with Spend Tracker

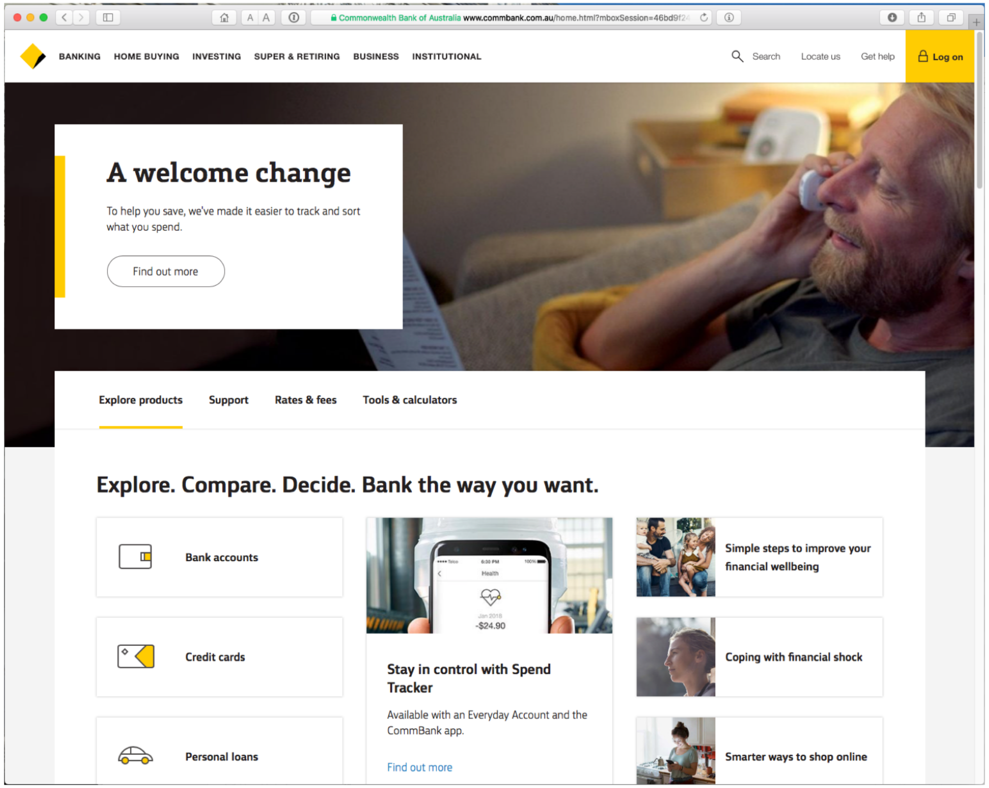
Available with an Everyday Account and the CommBank app.

Find out more
Simple steps to improve your financial wellbeing
Coping with financial shock
Smarter ways to shop online

I hope!

Security on the Internet

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



Commonwealth Bank of Australia www.commbank.com.au/home.html?mboxSession=46bd9f24

A welcome change

To help you save, we've made it easier to track and sort what you spend.

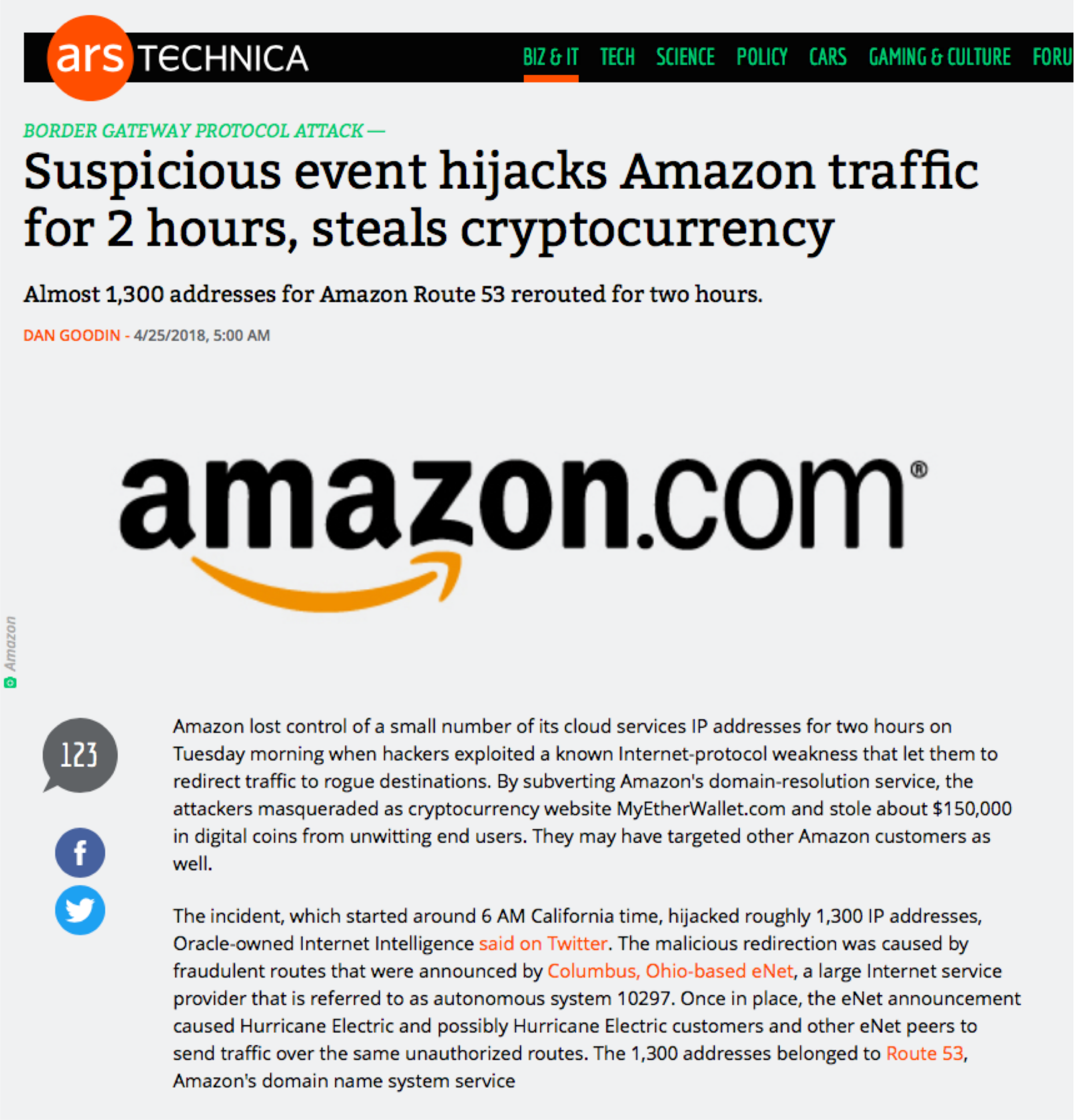
Find out more

Explore. Compare. Decide. Bank the way you want.

Stay in control with Spend Tracker

Available with an Everyday Account and the CommBank app.

Find out more



BORDER GATEWAY PROTOCOL ATTACK —

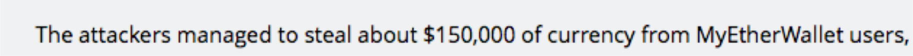
Suspicious event hijacks Amazon traffic for 2 hours, steals cryptocurrency

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

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

Amazon 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.

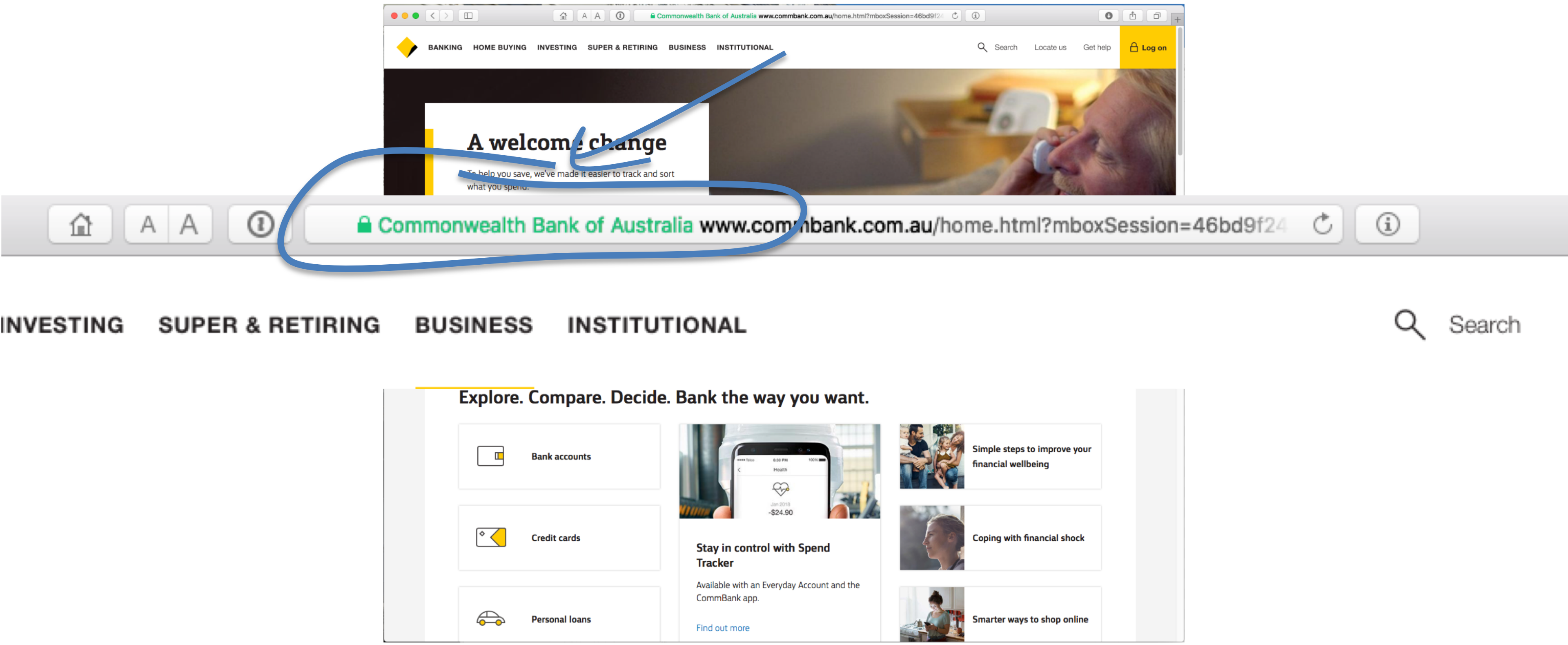
The incident, which started around 6 AM California time, hijacked roughly 1,300 IP addresses, Oracle-owned Internet Intelligence said on Twitter. The malicious redirection was caused by fraudulent routes that were announced by Columbus, Ohio-based eNet, a large Internet service provider that is referred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

Security on the Internet

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



Commonwealth Bank of Australia www.commbank.com.au/home.html?mboxSession=46bd9f24

A welcome change

To help you save, we've made it easier to track and sort what you spend.

Explore. Compare. Decide. Bank the way you want.

Stay in control with Spend Tracker

Available with an Everyday Account and the CommBank app.

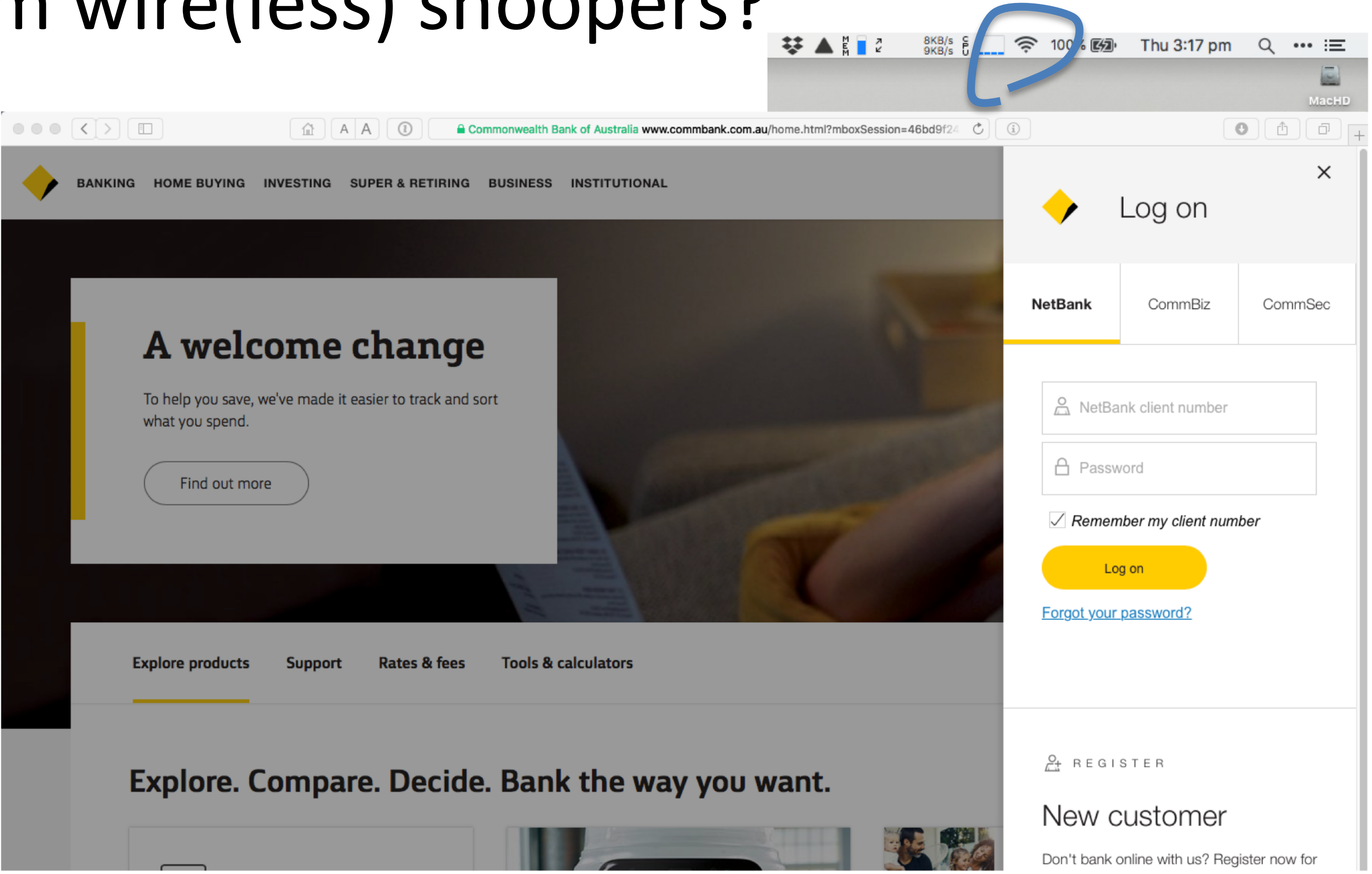
Simple steps to improve your financial wellbeing

Coping with financial shock

Smarter ways to shop online

Security on the Internet

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



8KB/s 9KB/s

100% Thu 3:17 pm

MacHD

Commonwealth Bank of Australia www.commbank.com.au/home.html?mboxSession=46bd9f24

BANKING HOME BUYING INVESTING SUPER & RETIRING BUSINESS INSTITUTIONAL

A welcome change

To help you save, we've made it easier to track and sort what you spend.

Find out more

Explore products Support Rates & fees Tools & calculators

Explore. Compare. Decide. Bank the way you want.

Log on

NetBank CommBiz CommSec

NetBank client number

Password

☑ *Remember my client number*

Log on

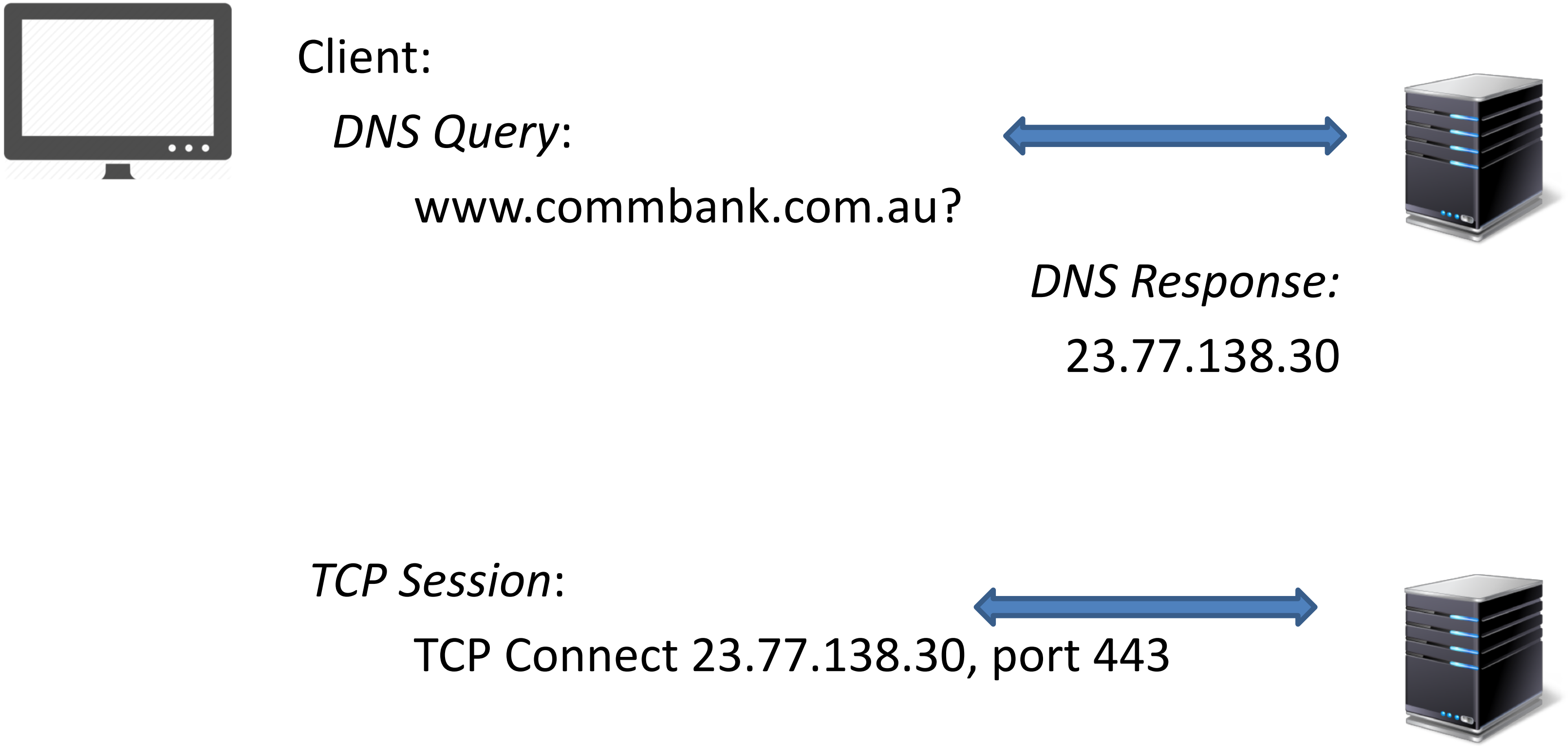
Forgot your password?

REGISTER

New customer

Don't bank online with us? Register now for

Opening the Connection: First Steps



Client:

DNS Query:

www.commbank.com.au?

DNS Response:

23.77.138.30

TCP Session:

TCP Connect 23.77.138.30, port 443

Hang on…

```
$ dig -x 23.77.138.30 +short
a23-77-138-30.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 23.77.138.30 +short
a23-77-138-30.deploy.static.akamaitechnologies.com.
```

That’s an Akamai address block

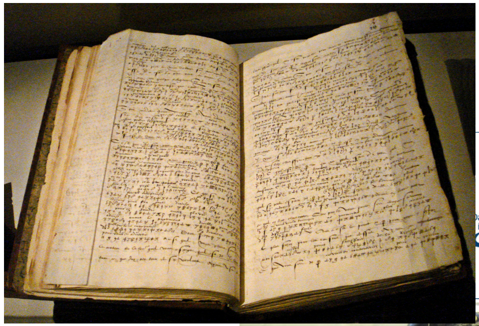
And I am NOT a customer of the Internet Bank of Akamai!

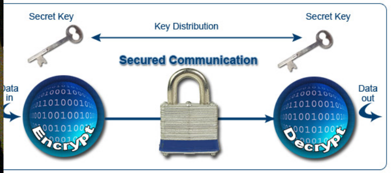
So why should my browser trust that 23.77.138.30 is really the “proper” web site for the Commonwealth Bank of Australia, and not some dastardly evil scam designed to steal my passwords and my money?

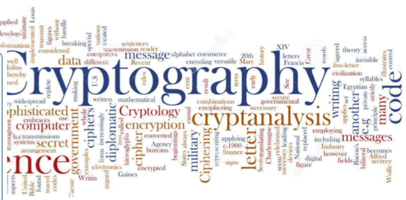
The major question…

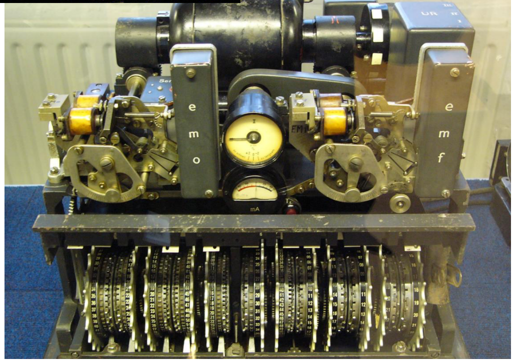
How does my browser tell the difference between an intended truth and a lie?

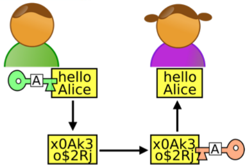
It’s all about cryptography











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

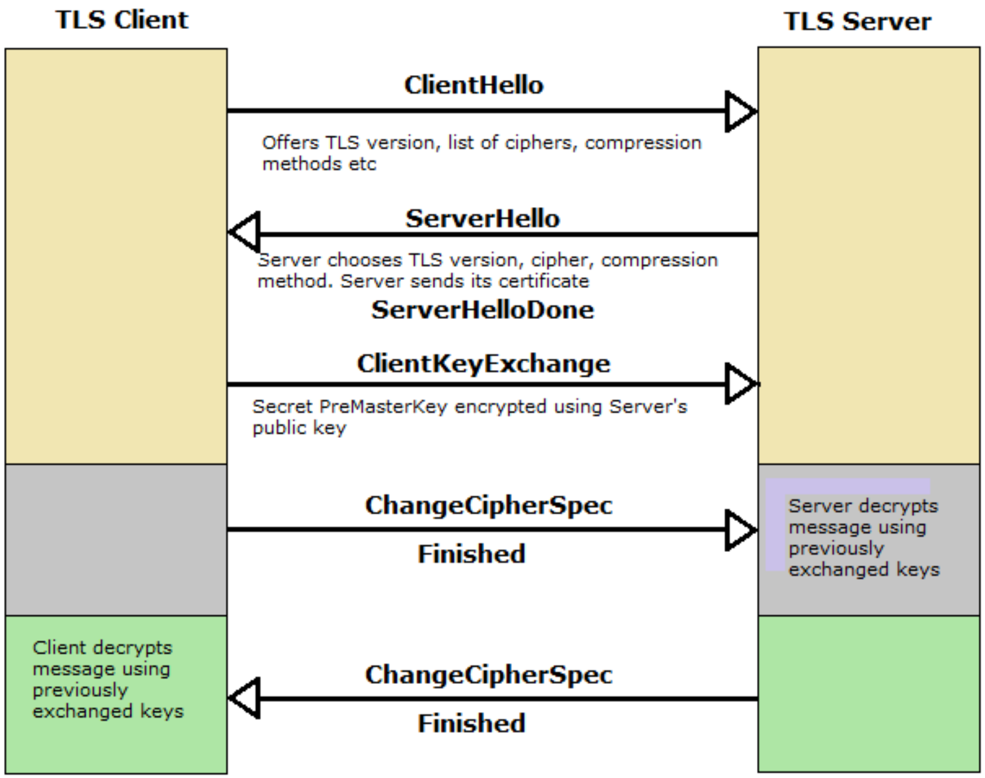
$$(m^e)^d \equiv m \pmod{n}$$

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?

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

Secure Connections using TLS



TLS Client

TLS Server

ClientHello

Offers TLS version, list of ciphers, compression methods etc

ServerHello

Server chooses TLS version, cipher, compression method. Server sends its certificate

ServerHelloDone

ClientKeyExchange

Secret PreMasterKey encrypted using Server's public key

ChangeCipherSpec

Finished

Server decrypts message using previously exchanged keys

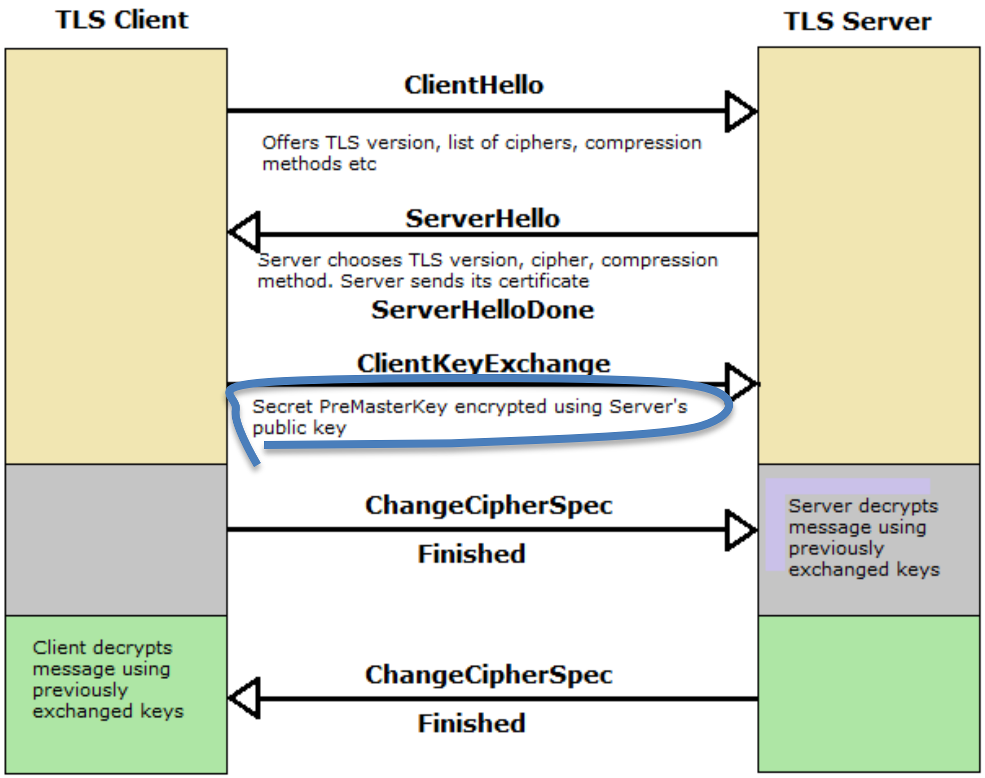
Client decrypts message using previously exchanged keys

ChangeCipherSpec

Finished

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

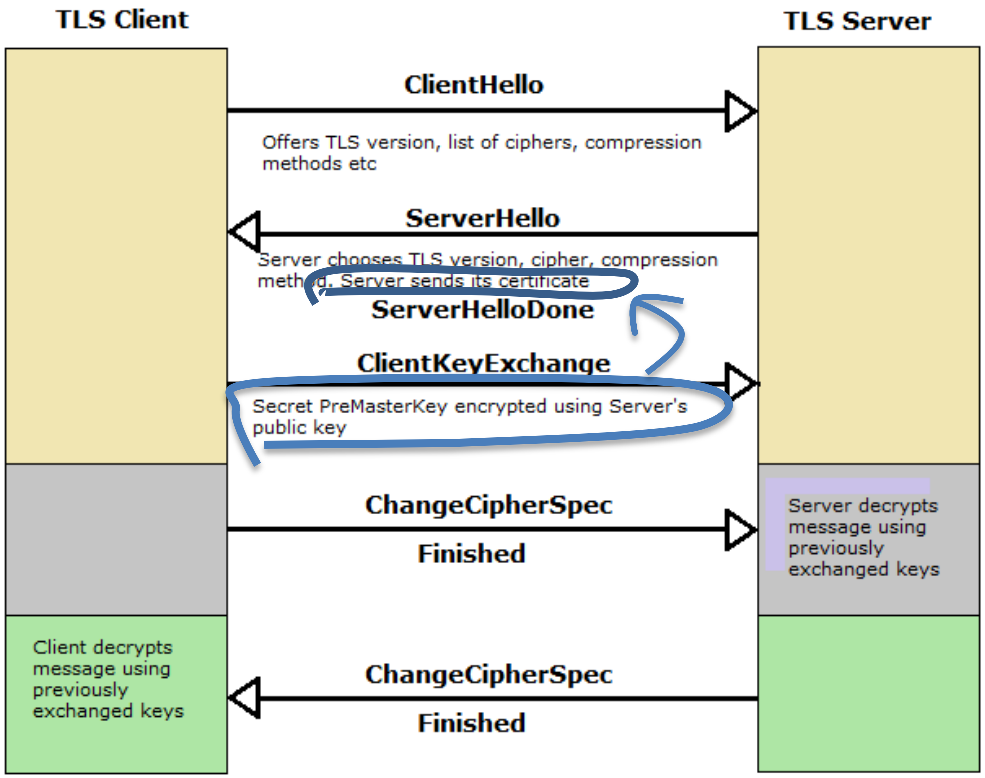
Secure Connections using TLS



TLS Client
TLS Server
ClientHello
Offers TLS version, list of ciphers, compression methods etc
ServerHello
Server chooses TLS version, cipher, compression method. Server sends its certificate
ServerHelloDone
ClientKeyExchange
Secret PreMasterKey encrypted using Server's public key
ChangeCipherSpec
Finished
Server decrypts message using previously exchanged keys
Client decrypts message using previously exchanged keys
ChangeCipherSpec
Finished

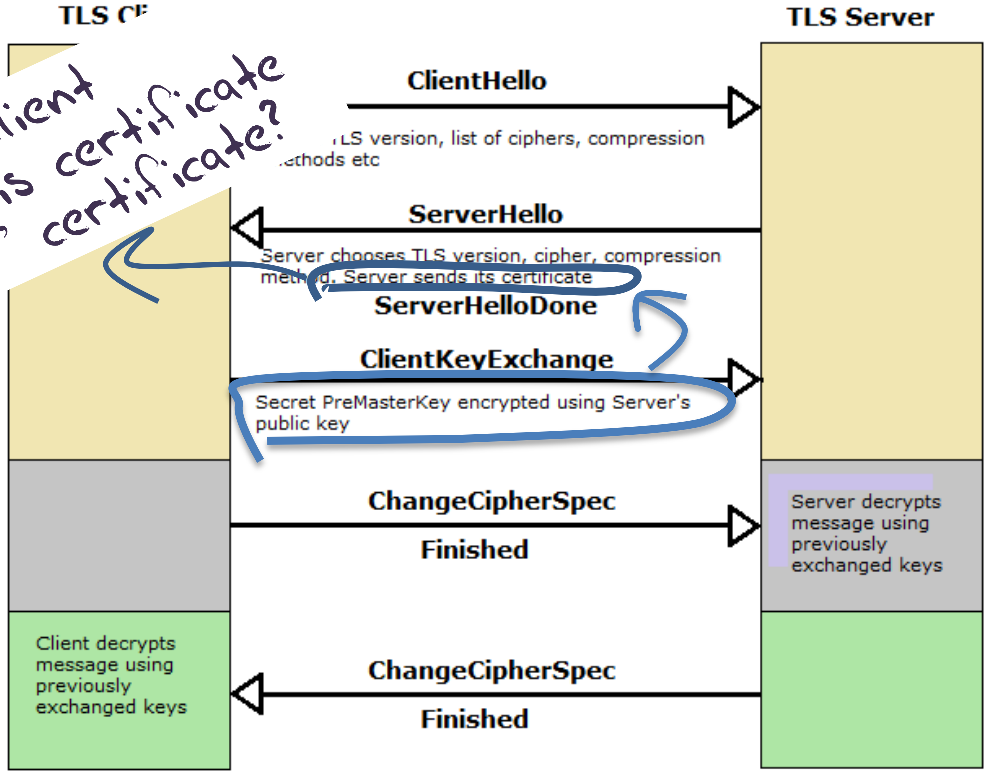
https://rhsecurity.wordpress.com/tag/tls/

Secure Connections using TLS


TLS Client
TLS Server
ClientHello
Offers TLS version, list of ciphers, compression methods etc
ServerHello
Server chooses TLS version, cipher, compression method. Server sends its certificate
ServerHelloDone
ClientKeyExchange
Secret PreMasterKey encrypted using Server's public key
ChangeCipherSpec
Finished
Server decrypts message using previously exchanged keys
Client decrypts message using previously exchanged keys
ChangeCipherSpec
Finished

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

Secure Connections using TLS



TLS C

TLS Server

ClientHello

LS version, list of ciphers, compression
ethods etc

ServerHello

Server chooses TLS version, cipher, compression method. Server sends its certificate

ServerHelloDone

ClientKeyExchange

Secret PreMasterKey encrypted using Server's public key

ChangeCipherSpec

Finished

Server decrypts message using previously exchanged keys

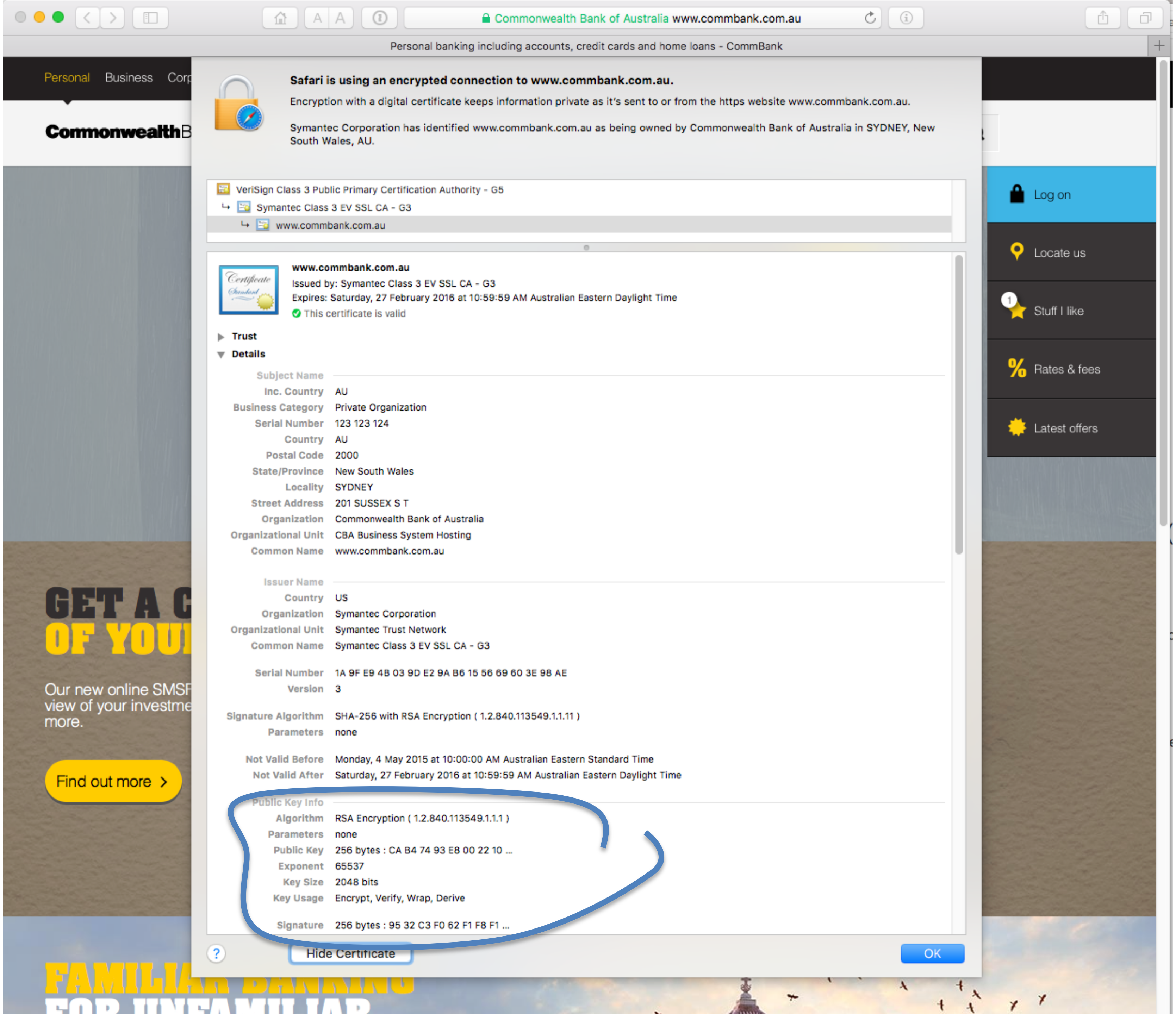
Client decrypts message using previously exchanged keys

ChangeCipherSpec

Finished

How does the client "recognise" this certificate as the "right" certificate?

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.

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

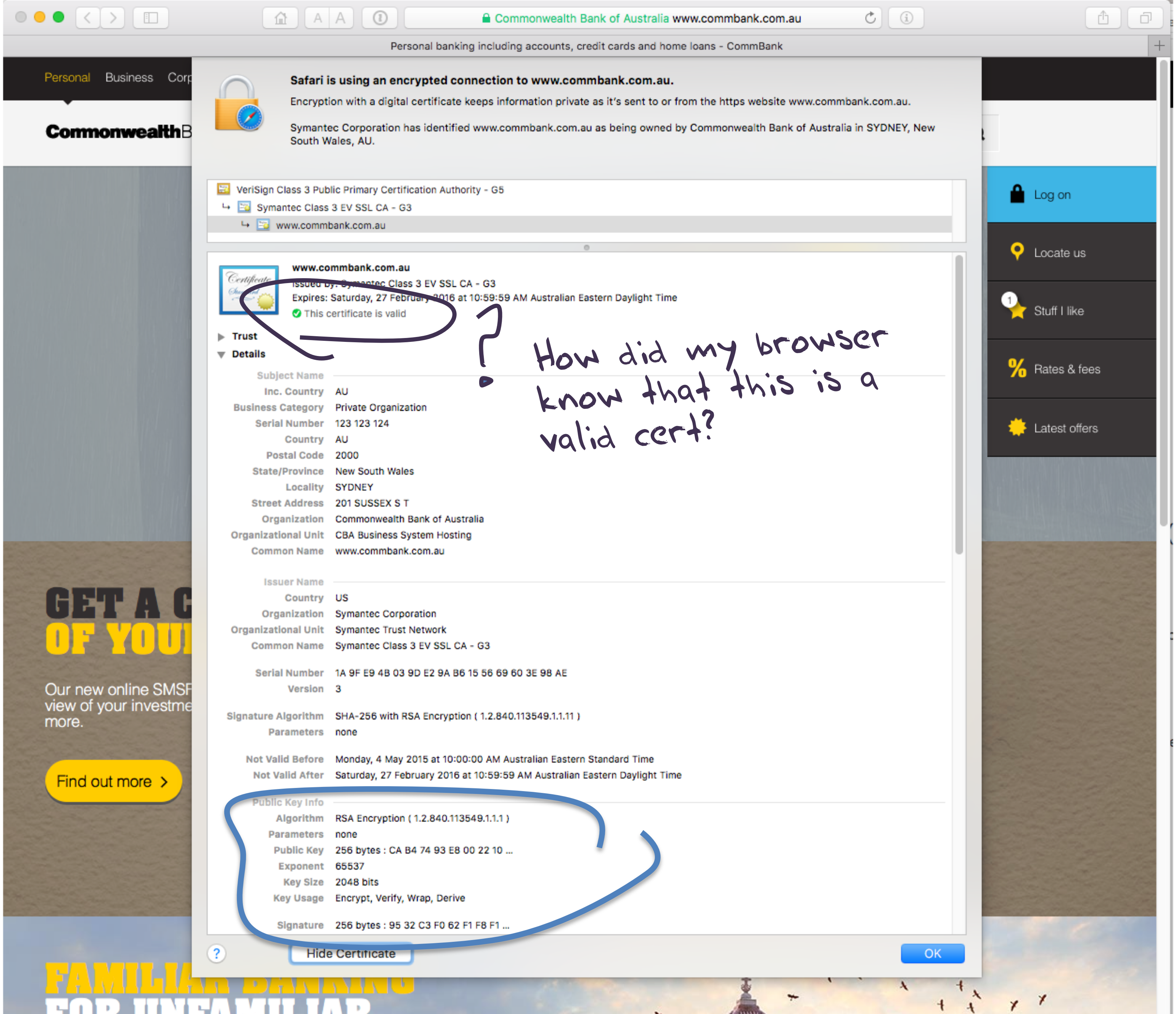
VeriSign Class 3 Public Primary Certification Authority - G5
↳ Symantec Class 3 EV SSL CA - G3
↳ www.commbank.com.au

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 ...



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 ...

?

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

Domain Name Certification

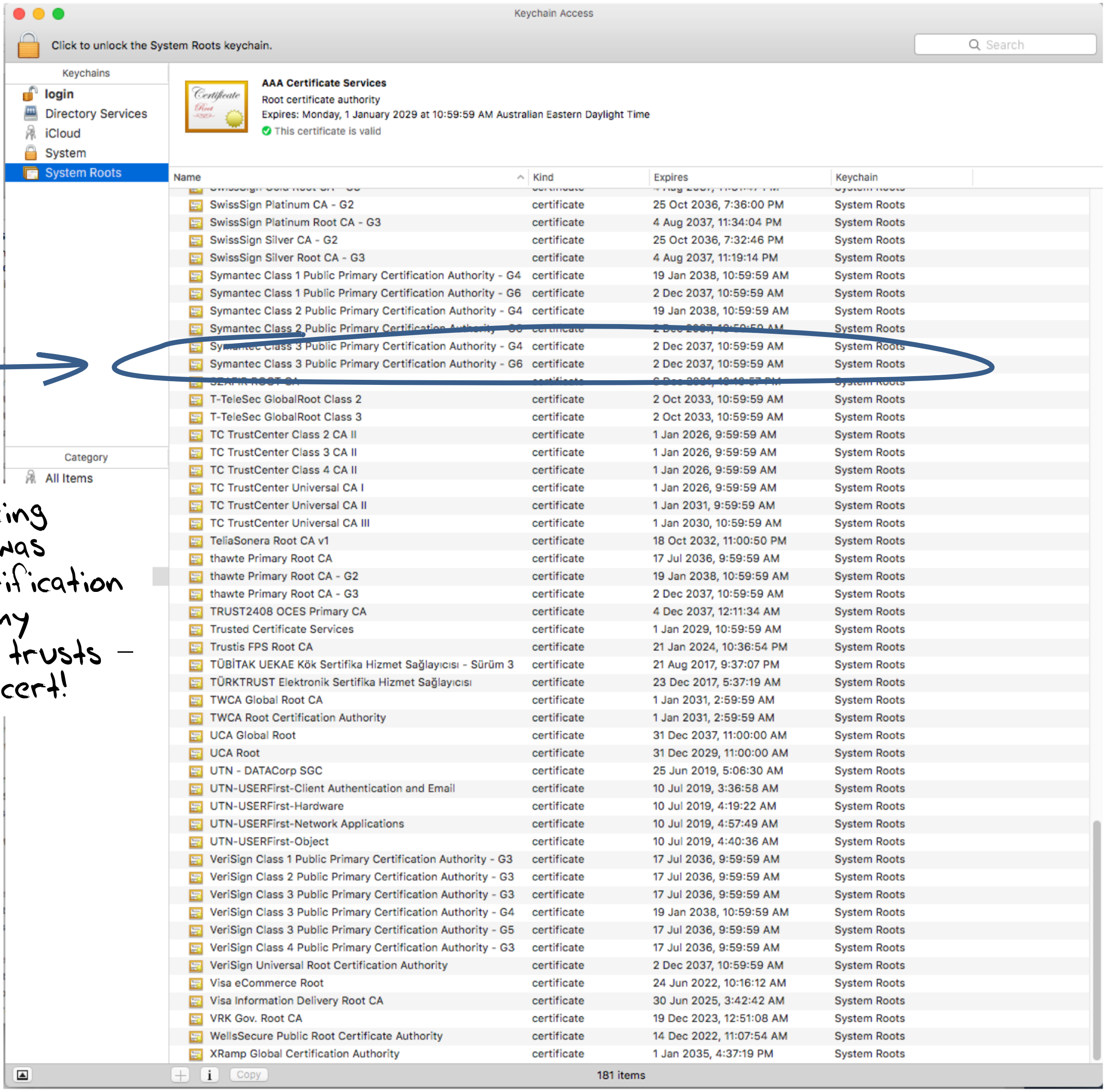
- The Commonwealth Bank of Australia has generated a key pair
- And they passed a certificate signing request to a company called “Symantec”
- Who was willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value
- So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to 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 Symantec and the certificates that they issue
- Symantec 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 “Symantec”
- Who was 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 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 Symantec and the certificates that they issue
- Symantec NEVER lie!

Why should i trust them?

Local Trust



AAA Certificate Services
Root certificate authority
Expires: Monday, 1 January 2029 at 10:59:59 AM Australian Eastern Daylight Time
✓ This certificate is valid

Name	Kind	Expires	Keychain
SwissSign Gold Root CA - G3	certificate	4 Aug 2037, [illegible]	System Roots
SwissSign Platinum CA - G2	certificate	25 Oct 2036, 7:36:00 PM	System Roots
SwissSign Platinum Root CA - G3	certificate	4 Aug 2037, 11:34:04 PM	System Roots
SwissSign Silver CA - G2	certificate	25 Oct 2036, 7:32:46 PM	System Roots
SwissSign Silver Root CA - G3	certificate	4 Aug 2037, 11:19:14 PM	System Roots
Symantec Class 1 Public Primary Certification Authority - G4	certificate	19 Jan 2038, 10:59:59 AM	System Roots
Symantec Class 1 Public Primary Certification Authority - G6	certificate	2 Dec 2037, 10:59:59 AM	System Roots
Symantec Class 2 Public Primary Certification Authority - G4	certificate	19 Jan 2038, 10:59:59 AM	System Roots
Symantec Class 2 Public Primary Certification Authority - G6	certificate	2 Dec 2037, 10:59:59 AM	System Roots
Symantec Class 3 Public Primary Certification Authority - G4	certificate	2 Dec 2037, 10:59:59 AM	System Roots
Symantec Class 3 Public Primary Certification Authority - G6	certificate	2 Dec 2037, 10:59:59 AM	System Roots
SZAFIR ROOT CA	certificate	6 Dec 2031, 10:10:57 PM	System Roots
T-TeleSec GlobalRoot Class 2	certificate	2 Oct 2033, 10:59:59 AM	System Roots
T-TeleSec GlobalRoot Class 3	certificate	2 Oct 2033, 10:59:59 AM	System Roots
TC TrustCenter Class 2 CA II	certificate	1 Jan 2026, 9:59:59 AM	System Roots
TC TrustCenter Class 3 CA II	certificate	1 Jan 2026, 9:59:59 AM	System Roots
TC TrustCenter Class 4 CA II	certificate	1 Jan 2026, 9:59:59 AM	System Roots
TC TrustCenter Universal CA I	certificate	1 Jan 2026, 9:59:59 AM	System Roots
TC TrustCenter Universal CA II	certificate	1 Jan 2031, 9:59:59 AM	System Roots
TC TrustCenter Universal CA III	certificate	1 Jan 2030, 10:59:59 AM	System Roots
TeliaSonera Root CA v1	certificate	18 Oct 2032, 11:00:50 PM	System Roots
thawte Primary Root CA	certificate	17 Jul 2036, 9:59:59 AM	System Roots
thawte Primary Root CA - G2	certificate	19 Jan 2038, 10:59:59 AM	System Roots
thawte Primary Root CA - G3	certificate	2 Dec 2037, 10:59:59 AM	System Roots
TRUST2408 OCES Primary CA	certificate	4 Dec 2037, 12:11:34 AM	System Roots
Trusted Certificate Services	certificate	1 Jan 2029, 10:59:59 AM	System Roots
Trustis FPS Root CA	certificate	21 Jan 2024, 10:36:54 PM	System Roots
TÜBİTAK UEKAE Kök Sertifika Hizmet Sağlayıcısı - Sürüm 3	certificate	21 Aug 2017, 9:37:07 PM	System Roots
TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı	certificate	23 Dec 2017, 5:37:19 AM	System Roots
TWCA Global Root CA	certificate	1 Jan 2031, 2:59:59 AM	System Roots
TWCA Root Certification Authority	certificate	1 Jan 2031, 2:59:59 AM	System Roots
UCA Global Root	certificate	31 Dec 2037, 11:00:00 AM	System Roots
UCA Root	certificate	31 Dec 2029, 11:00:00 AM	System Roots
UTN - DATACorp SGC	certificate	25 Jun 2019, 5:06:30 AM	System Roots
UTN-USERFirst-Client Authentication and Email	certificate	10 Jul 2019, 3:36:58 AM	System Roots
UTN-USERFirst-Hardware	certificate	10 Jul 2019, 4:19:22 AM	System Roots
UTN-USERFirst-Network Applications	certificate	10 Jul 2019, 4:57:49 AM	System Roots
UTN-USERFirst-Object	certificate	10 Jul 2019, 4:40:36 AM	System Roots
VeriSign Class 1 Public Primary Certification Authority - G3	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Class 2 Public Primary Certification Authority - G3	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Class 3 Public Primary Certification Authority - G3	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Class 3 Public Primary Certification Authority - G4	certificate	19 Jan 2038, 10:59:59 AM	System Roots
VeriSign Class 3 Public Primary Certification Authority - G5	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Class 4 Public Primary Certification Authority - G3	certificate	17 Jul 2036, 9:59:59 AM	System Roots
VeriSign Universal Root Certification Authority	certificate	2 Dec 2037, 10:59:59 AM	System Roots
Visa eCommerce Root	certificate	24 Jun 2022, 10:16:12 AM	System Roots
Visa Information Delivery Root CA	certificate	30 Jun 2025, 3:42:42 AM	System Roots
VRK Gov. Root CA	certificate	19 Dec 2023, 12:51:08 AM	System Roots
WellsSecure Public Root Certificate Authority	certificate	14 Dec 2022, 11:07:54 AM	System Roots
XRamp Global Certification Authority	certificate	1 Jan 2035, 4:37:19 PM	System Roots

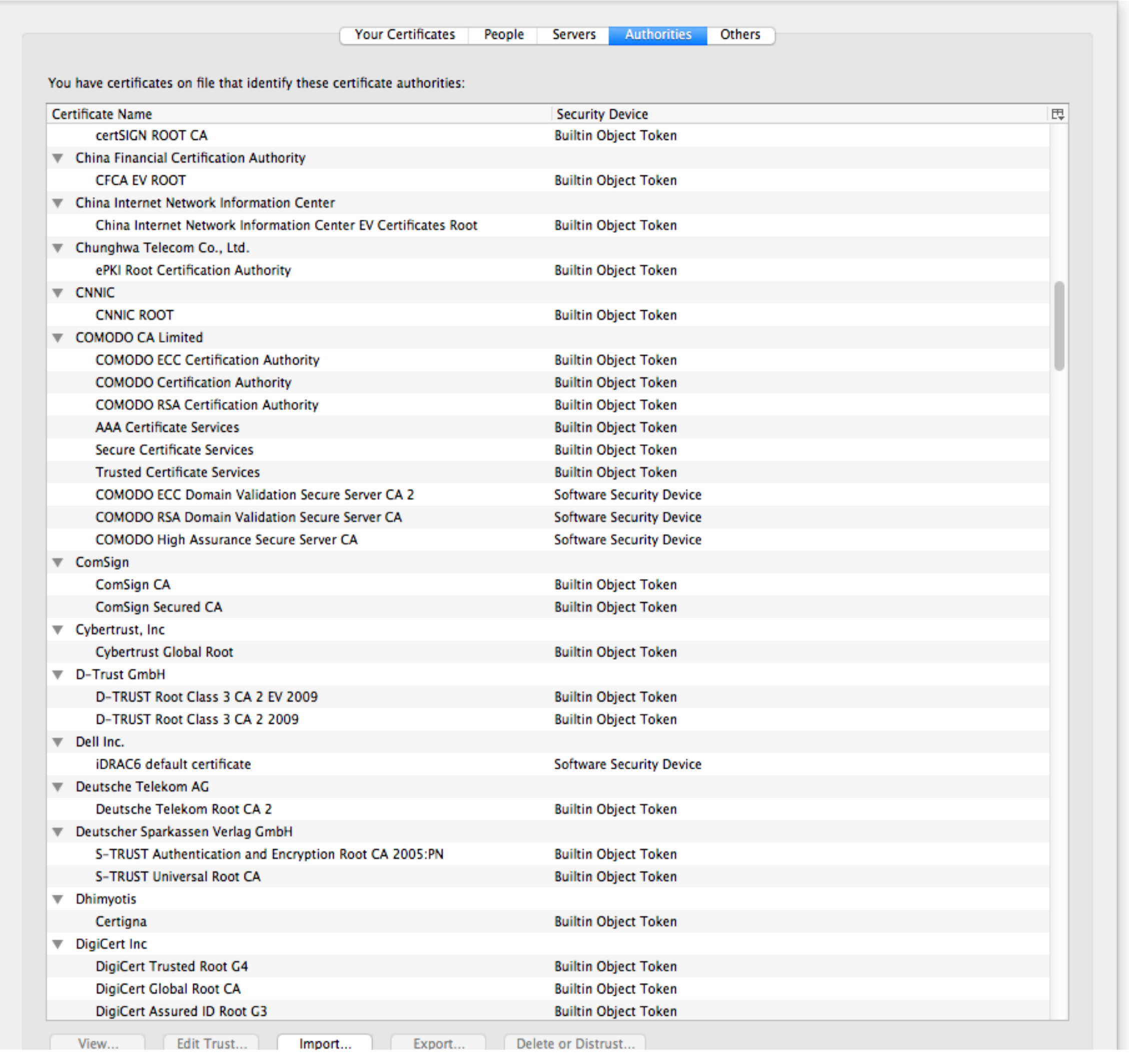
181 items

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?



You have certificates on file that identify these certificate authorities:

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

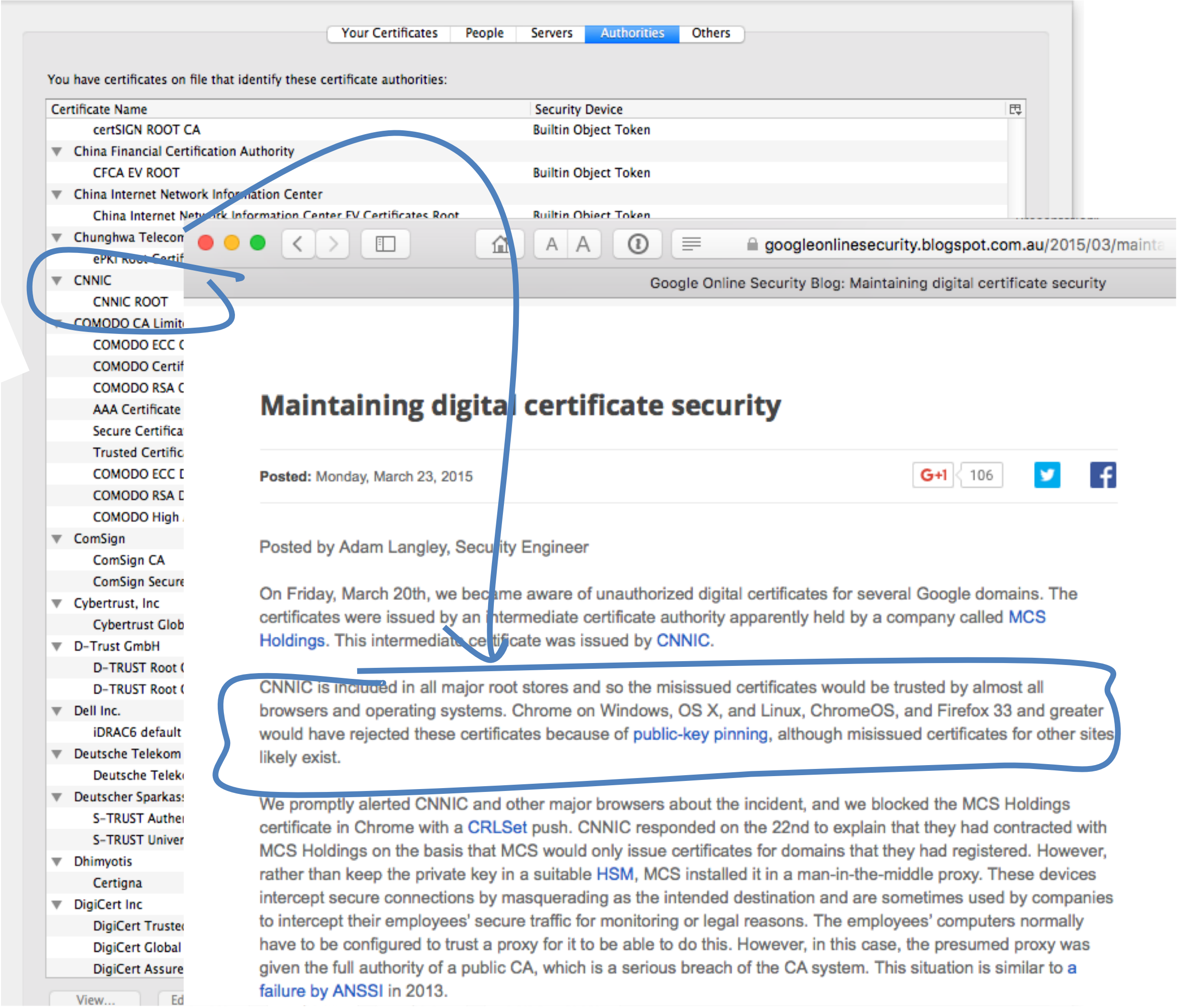
* cre·du·li·ty
/krəˈd(y)o͞olədē/
noun
a tendency to be too ready to believe that something is real or true.

Local Credulity

That's a big list of people to Trust

Are they all trustable?

Evidently Not!


Your Certificates
People
Servers
Authorities
Others
View...
Ed
googleonlinesecurity.blogspot.com.au/2015/03/mainta
Google Online Security Blog: Maintaining digital certificate security
G+1
106

You have certificates on file that identify these certificate authorities:

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	
ePKI Root Certif	
CNNIC	
CNNIC ROOT	
COMODO CA Limit	
COMODO ECC C	
COMODO Certif	
COMODO RSA C	
AAA Certificate	
Secure Certifica	
Trusted Certific	
COMODO ECC D	
COMODO RSA D	
COMODO High	
ComSign	
ComSign CA	
ComSign Secure	
Cybertrust, Inc	
Cybertrust Glob	
D-Trust GmbH	
D-TRUST Root C	
D-TRUST Root C	
Dell Inc.	
iDRAC6 default	
Deutsche Telekom	
Deutsche Telek	
Deutscher Sparkas	
S-TRUST Authen	
S-TRUST Univer	
Dhimyotis	
Certigna	
DigiCert Inc	
DigiCert Truste	
DigiCert Global	
DigiCert Assure	

Maintaining digital certificate security

Posted: Monday, March 23, 2015

Posted by Adam Langley, Security Engineer

On Friday, March 20th, we became aware of unauthorized digital certificates for several Google domains. The certificates were issued by an intermediate certificate authority apparently held by a company called MCS Holdings. This intermediate certificate was issued by CNNIC.

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

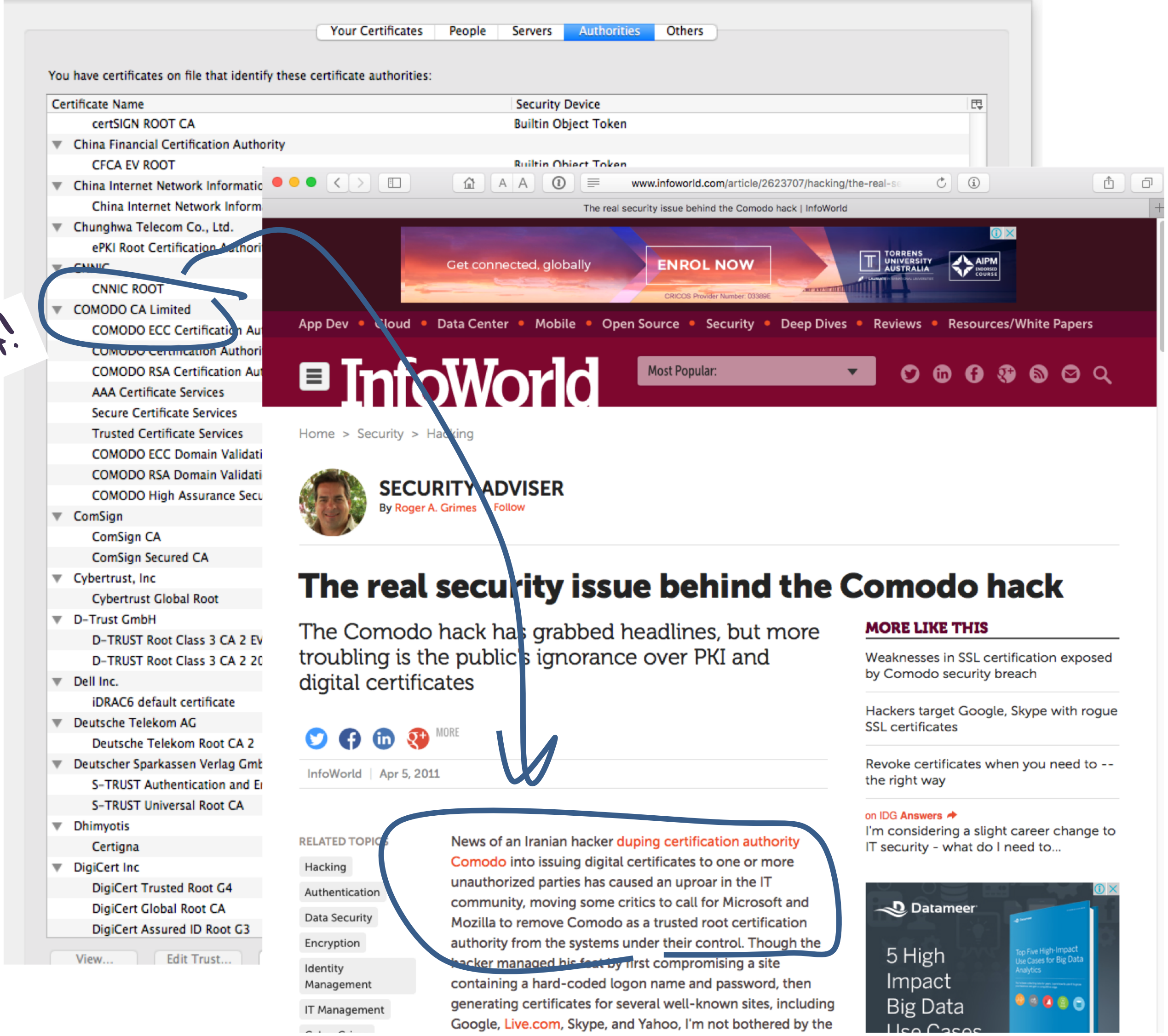
We promptly alerted CNNIC and other major browsers about the incident, and we blocked the MCS Holdings certificate in Chrome with a CRLSet push. CNNIC responded on the 22nd to explain that they had contracted with MCS Holdings on the basis that MCS would only issue certificates for domains that they had registered. However, rather than keep the private key in a suitable HSM, MCS installed it in a man-in-the-middle proxy. These devices intercept secure connections by masquerading as the intended destination and are sometimes used by companies to intercept their employees' secure traffic for monitoring or legal reasons. The employees' computers normally have to be configured to trust a proxy for it to be able to do this. However, in this case, the presumed proxy was given the full authority of a public CA, which is a serious breach of the CA system. This situation is similar to a failure by ANSSI in 2013.

Local Credulity

That's a big list of people to Trust

Are they all trustable?

Evidently Not!



You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
certSIGN ROOT CA	Builtin Object Token
China Financial Certification Authority	
CFCA EV ROOT	Builtin Object Token
China Internet Network Informatio	
China Internet Network Inform	
Chunghwa Telecom Co., Ltd.	
ePKI Root Certification Authori	
CNNIC	
CNNIC ROOT	
COMODO CA Limited	
COMODO ECC Certification Au	
COMODO Certification Authori	
COMODO RSA Certification Au	
AAA Certificate Services	
Secure Certificate Services	
Trusted Certificate Services	
COMODO ECC Domain Validati	
COMODO RSA Domain Validati	
COMODO High Assurance Secu	
ComSign	
ComSign CA	
ComSign Secured CA	
Cybertrust, Inc	
Cybertrust Global Root	
D-Trust GmbH	
D-TRUST Root Class 3 CA 2 EV	
D-TRUST Root Class 3 CA 2 20	
Dell Inc.	
iDRAC6 default certificate	
Deutsche Telekom AG	
Deutsche Telekom Root CA 2	
Deutscher Sparkassen Verlag Gmb	
S-TRUST Authentication and E	
S-TRUST Universal Root CA	
Dhimyotis	
Certigna	
DigiCert Inc	
DigiCert Trusted Root G4	
DigiCert Global Root CA	
DigiCert Assured ID Root G3	

www.infoworld.com/article/2623707/hacking/the-real-s

The real security issue behind the Comodo hack | InfoWorld

App Dev • Cloud • Data Center • Mobile • Open Source • Security • Deep Dives • Reviews • Resources/White Papers
InfoWorld
Most Popular:

Home > Security > Hacking

SECURITY ADVISER
By Roger A. Grimes Follow

The real security issue behind the Comodo hack

The Comodo hack has grabbed headlines, but more troubling is the public's ignorance over PKI and digital certificates

InfoWorld | Apr 5, 2011

RELATED TOPICS

News of an Iranian hacker duping certification authority Comodo into issuing digital certificates to one or more unauthorized parties has caused an uproar in the IT community, moving some critics to call for Microsoft and Mozilla to remove Comodo as a trusted root certification authority from the systems under their control. Though the hacker managed his feat by first compromising a site containing a hard-coded logon name and password, then generating certificates for several well-known sites, including Google, Live.com, Skype, and Yahoo, I'm not bothered by the

MORE LIKE THIS

Weaknesses in SSL certification exposed by Comodo security breach

Hackers target Google, Skype with rogue SSL certificates

Revoke certificates when you need to -- the right way

on IDG Answers
I'm considering a slight career change to IT security - what do I need to...

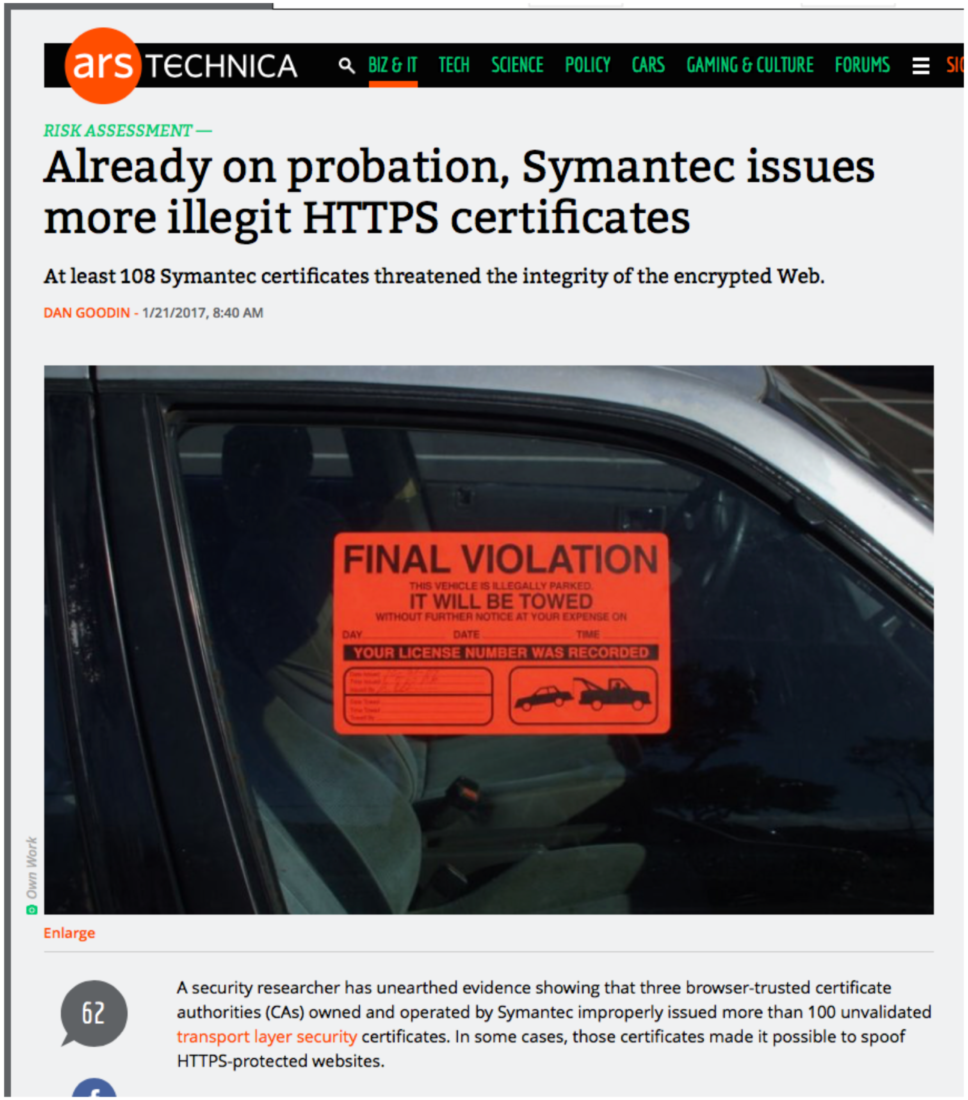
But my bank used Symantec

as their Certificate Authority

And Symantec NEVER lie in the certificates they issue

Never?

Well, hardly ever



RISK ASSESSMENT —

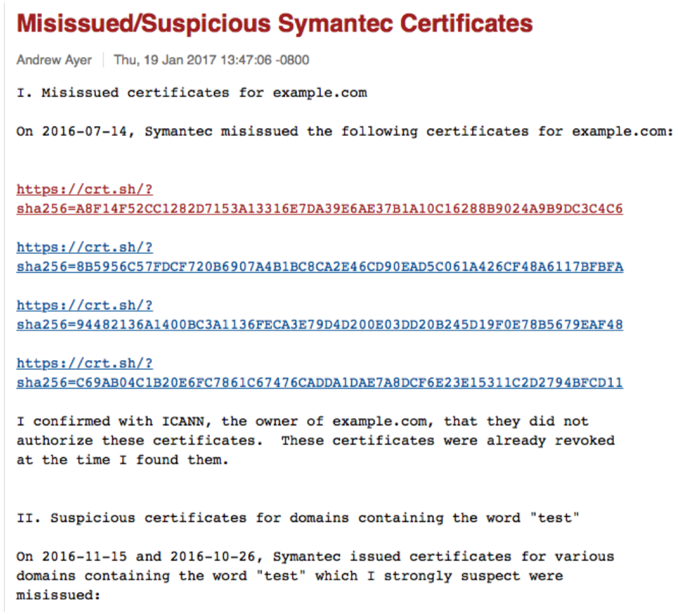
Already on probation, Symantec issues more illegit HTTPS certificates

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

DAN GOODIN - 1/21/2017, 8:40 AM

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/

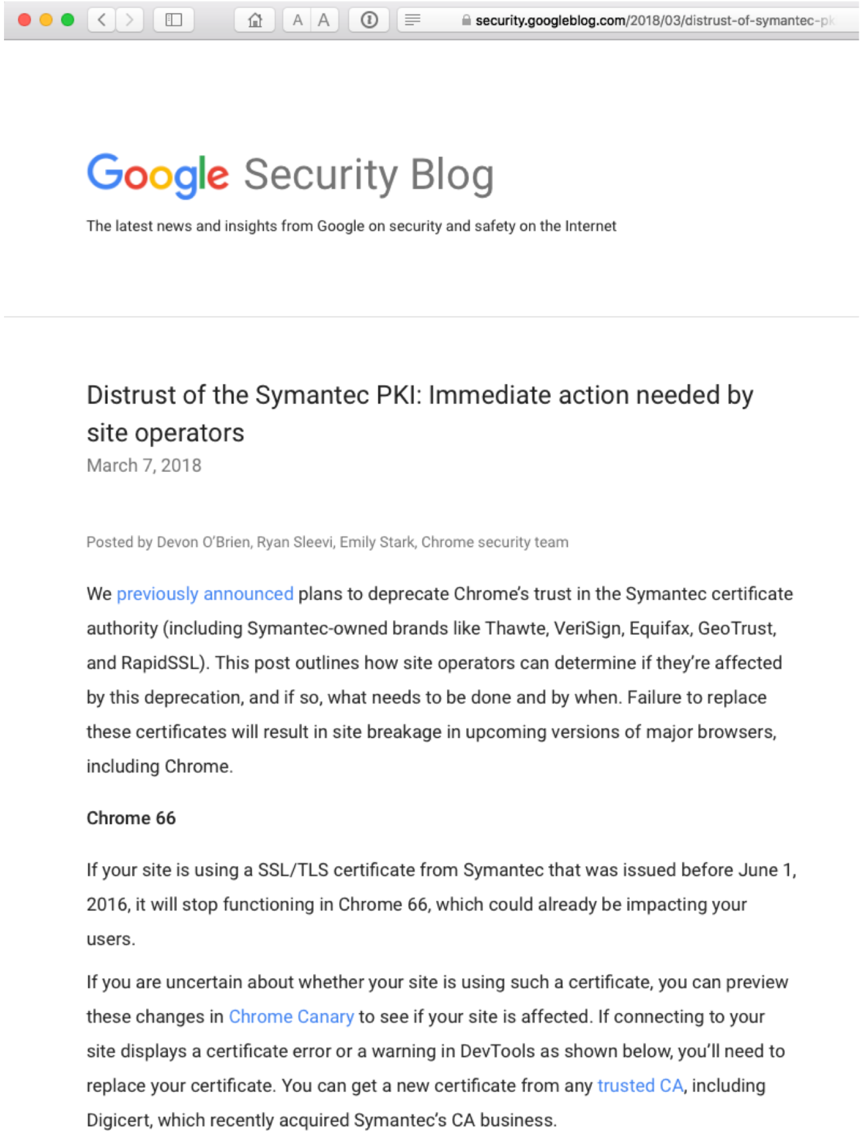


Misissued/Suspicious Symantec Certificates

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



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.

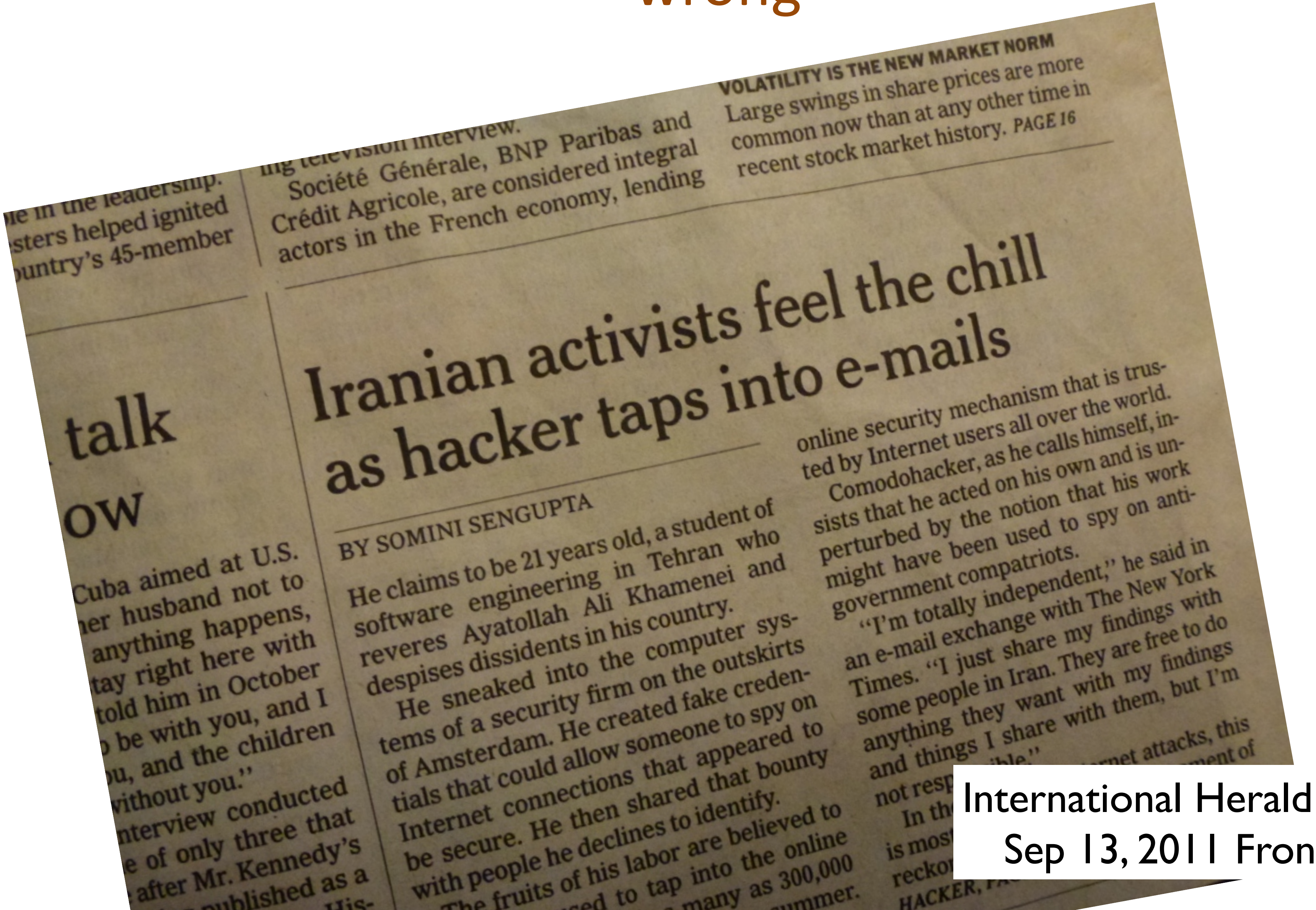
Chrome 66

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



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*

ple in the leadership.
esters helped ignited
ountry's 45-member

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
ou, and the children
without you."
nterview conducted
e of only three that
after Mr. Kennedy's
published as a
His-

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 [illegible] used to tap into the online [illegible] many as 300,000 [illegible] summer.

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 antigovernment 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[illegible]ible."

In th[illegible]ernet attacks, this is most[illegible]ment of reckon[illegible]

HACKER, PA[illegible]

International Herald Tribune
Sep 13, 2011 Front Page

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 … dig … 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 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?

Sustainable
Resilient
Secure
Privacy
Trusted

?

In a commercial environment

Where CA’s compete with each other for market share

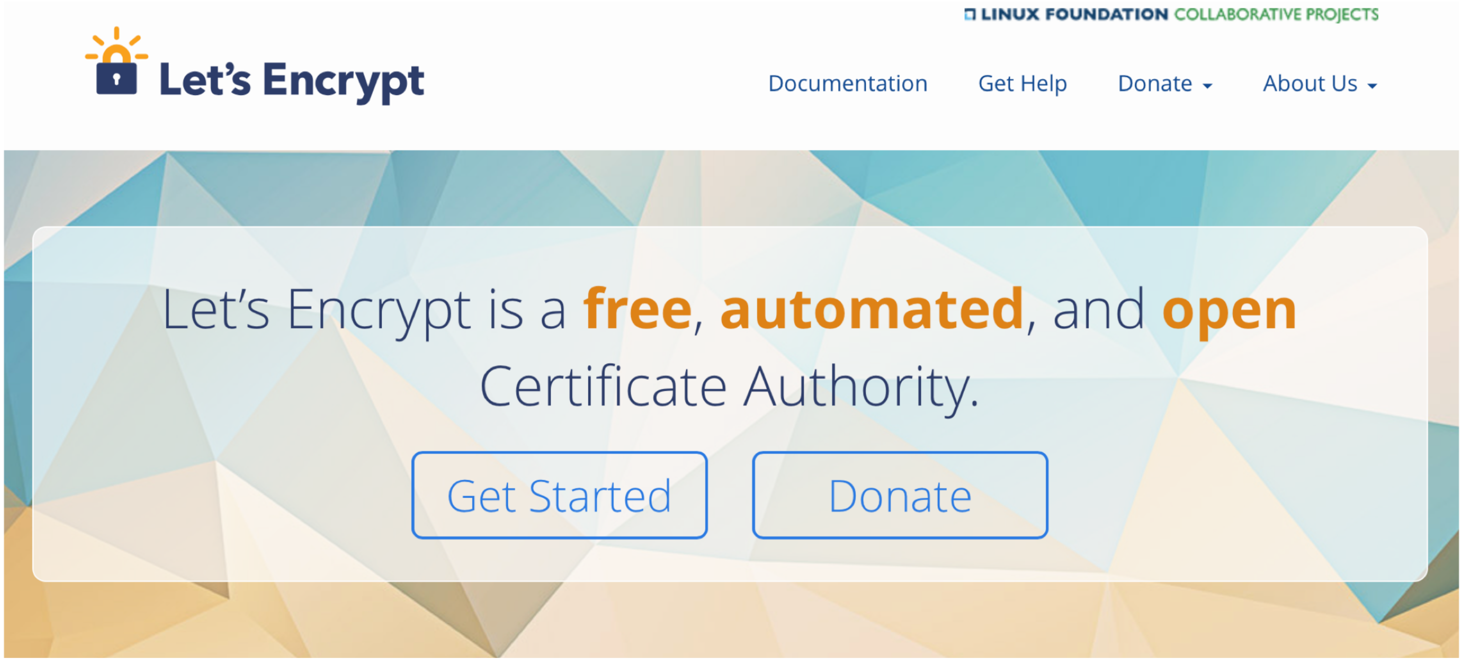
And quality offers no protection

Than what ‘wins’ in the market?

Sustainable
Resilient
Secure
Privacy
Trusted
Cheap!

Where now?

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



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

Where now?

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

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

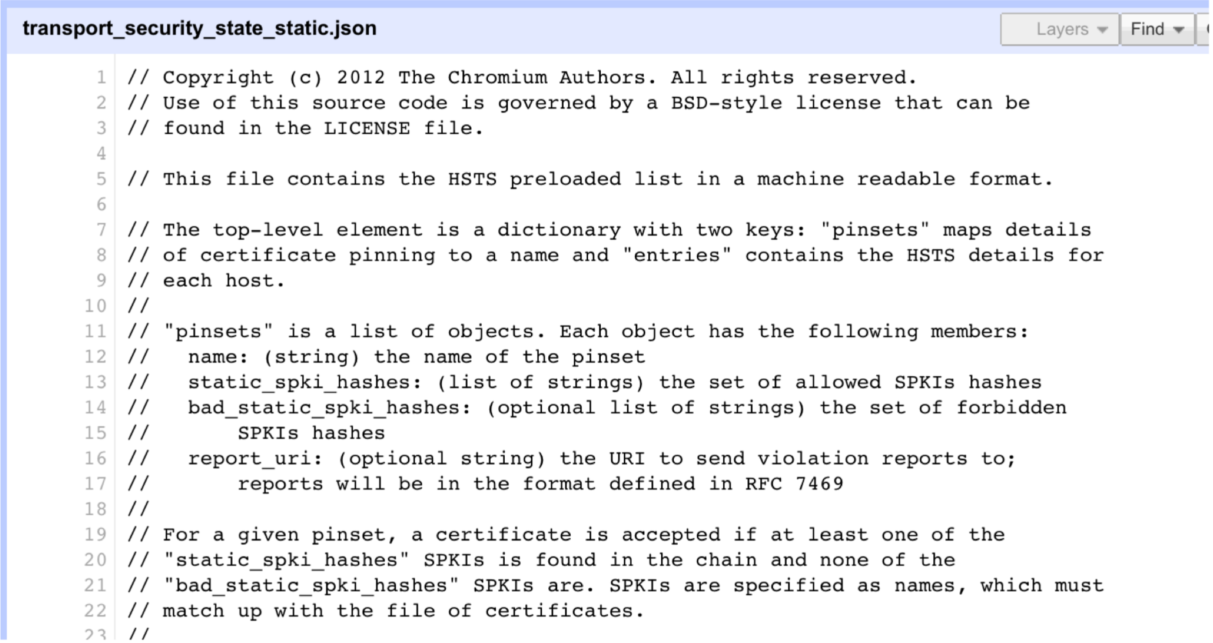
Will the automation of the Cert issuance coupled with a totally free service make the overall environment more or less secure?

We're probably going to find out real soon!

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



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.
//
```

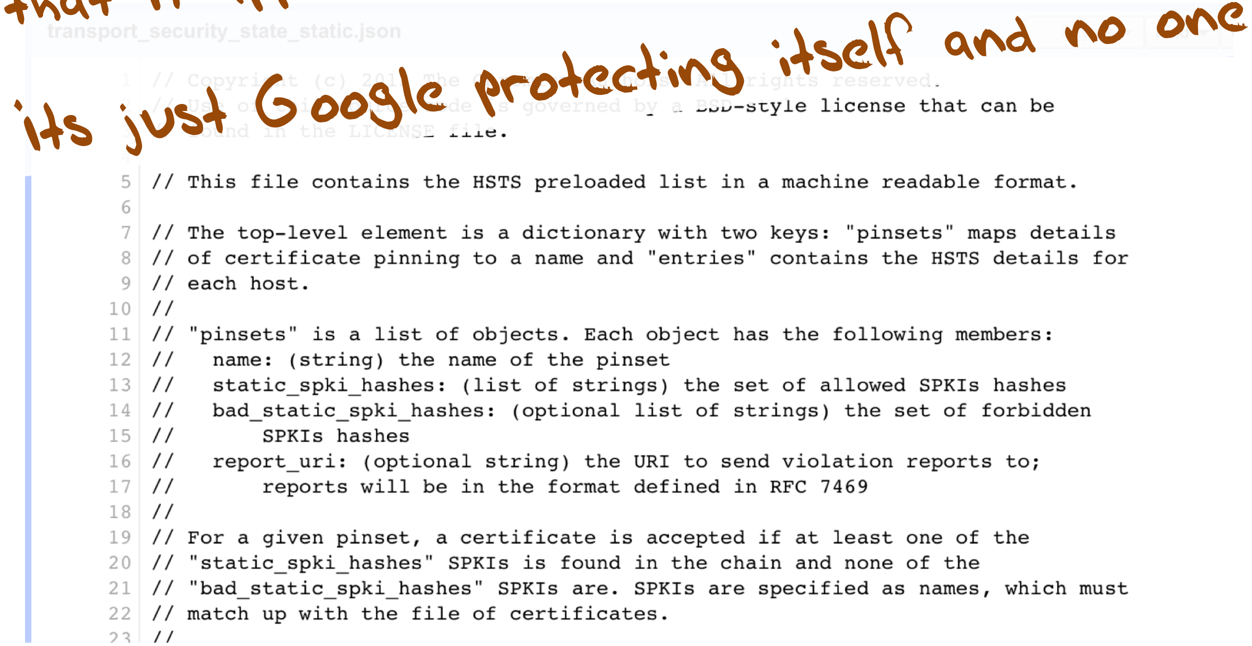
Where now?

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!

its just Google protecting itself and no one else



transport_security_state_static.json

```
// Copyright (c) 201[illegible] 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.
//
```

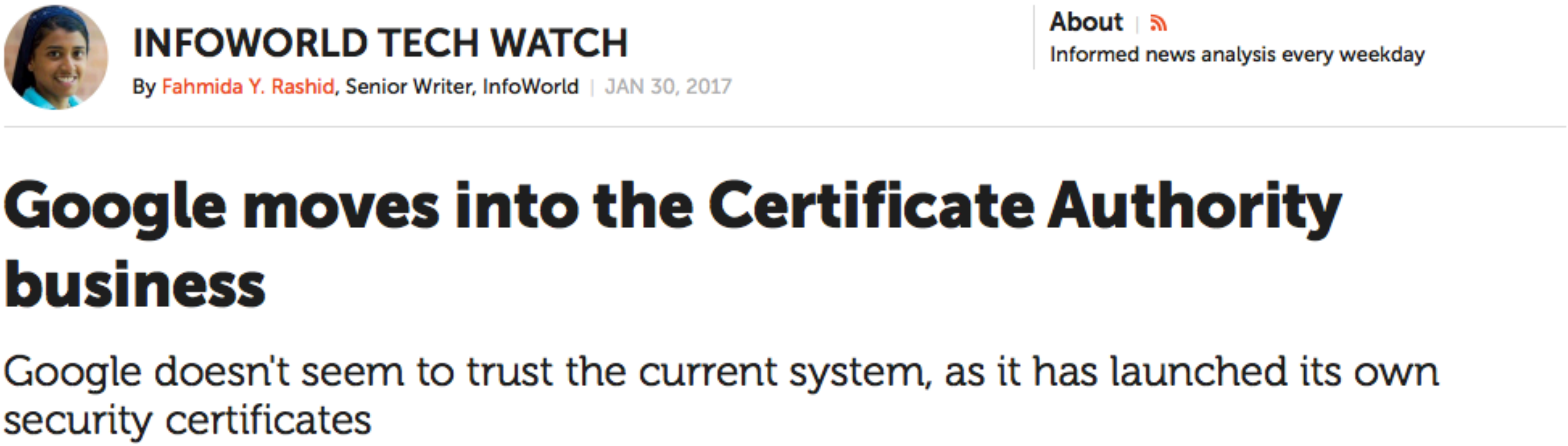
Where now?

Option B: White Listing and Pinning with HSTS

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

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



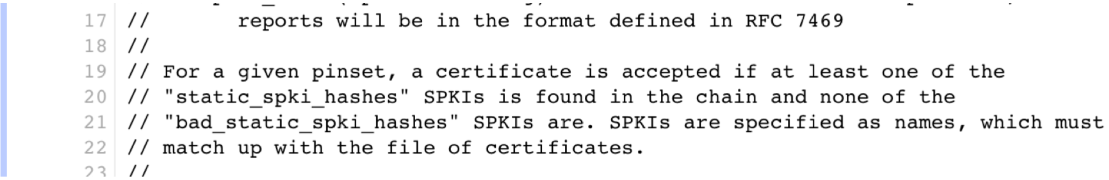
INFOWORLD TECH WATCH

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



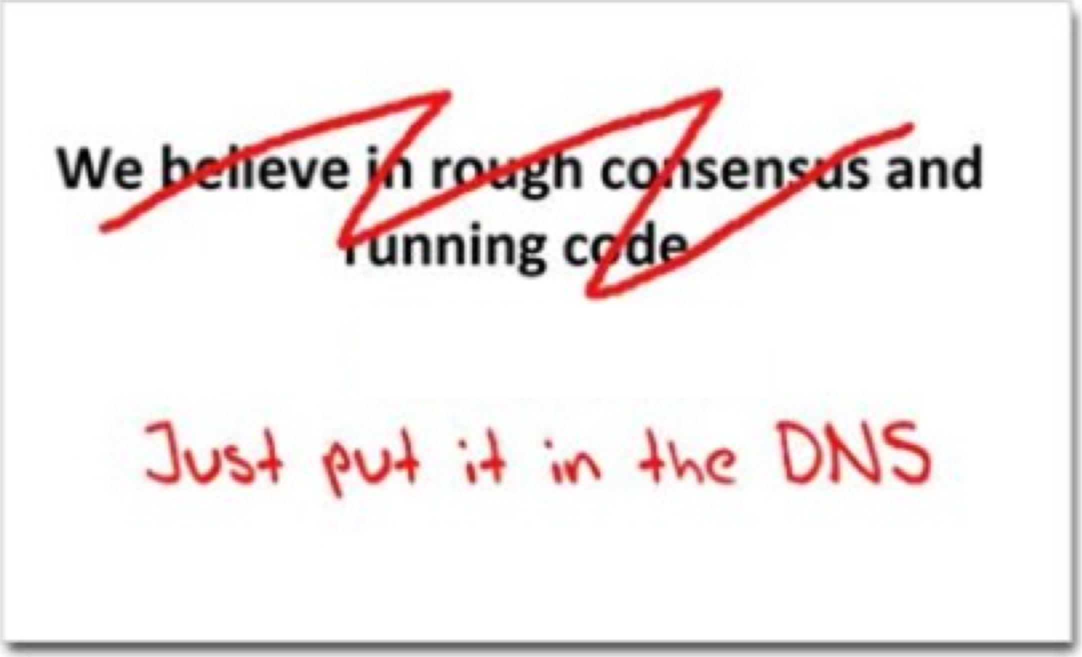
```
//       reports will be in the format defined in RFC 7469
//
// For a given pinset, a certificate is accepted if at least one of the
// "static_spki_hashes" SPKIs is found in the chain and none of the
// "bad_static_spki_hashes" SPKIs are. SPKIs are specified as names, which must
// match up with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

Where now?

Option C: Use the DNS!

Where now?

Option C: Use the DNS!



~~We believe in rough consensus and running code~~

Just put it in 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?

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 (pinning record)?

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?

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?

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 th[illegible] the HSTS record?
- Why not [illegible] DNS for the issuer CA?
- W[illegible] query the DNS for the hash of the [illegible]ain name cert?
- Why not query the DNS for the hash of the domain name public key?

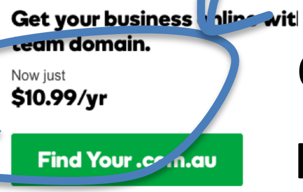
Who needs CA's anyway?

Seriously ? The DNS?

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

- Why not query th . the HSTS record?
- Why not DNS for the issuer CA?
- W query the DNS fo ain name cert?
- ot query the DNS fo n name public key?

Who needs CA's anyway?

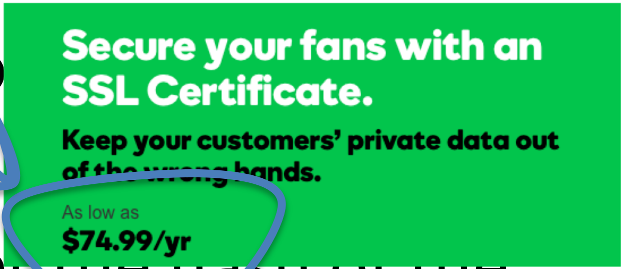


Get your business online wit team domain.

Now just

$10.99/yr

Find Your .com.au



Secure your fans with an SSL Certificate.

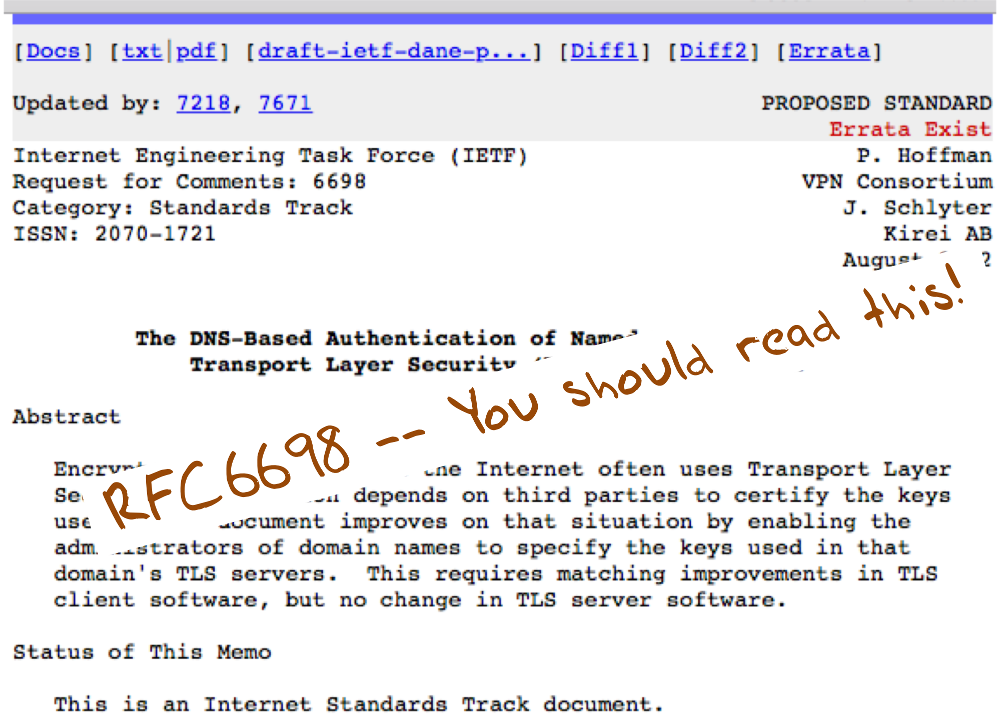
Keep your customers' private data out of the wrong hands.

As low as

$74.99/yr

DANE

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

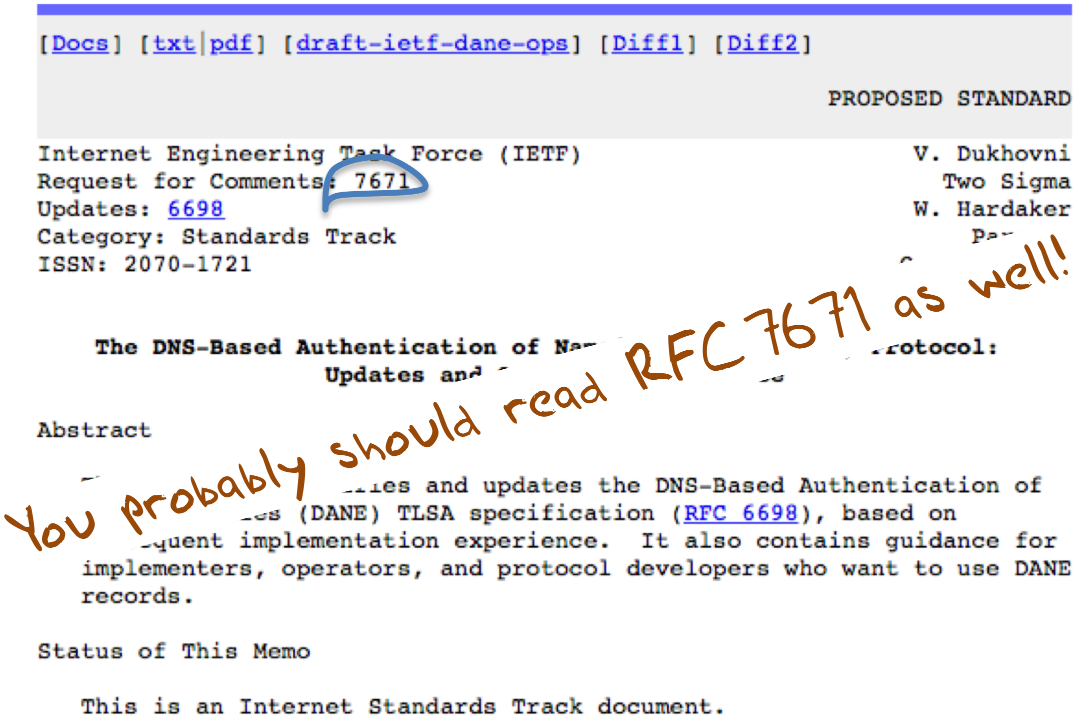


```
[Docs] [txt|pdf] [draft-ietf-dane-p...] [Diff1] [Diff2] [Errata]

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
                                                               Augus   2


         The DNS-Based Authentication of Name
             Transport Layer Security

Abstract

   Encry                 he Internet often uses Transport Layer
   Se                   n depends on third parties to certify the keys
   use            document improves on that situation by enabling the
   adm  strators 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.
```

RFC6698 -- You should read this!

DANE

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

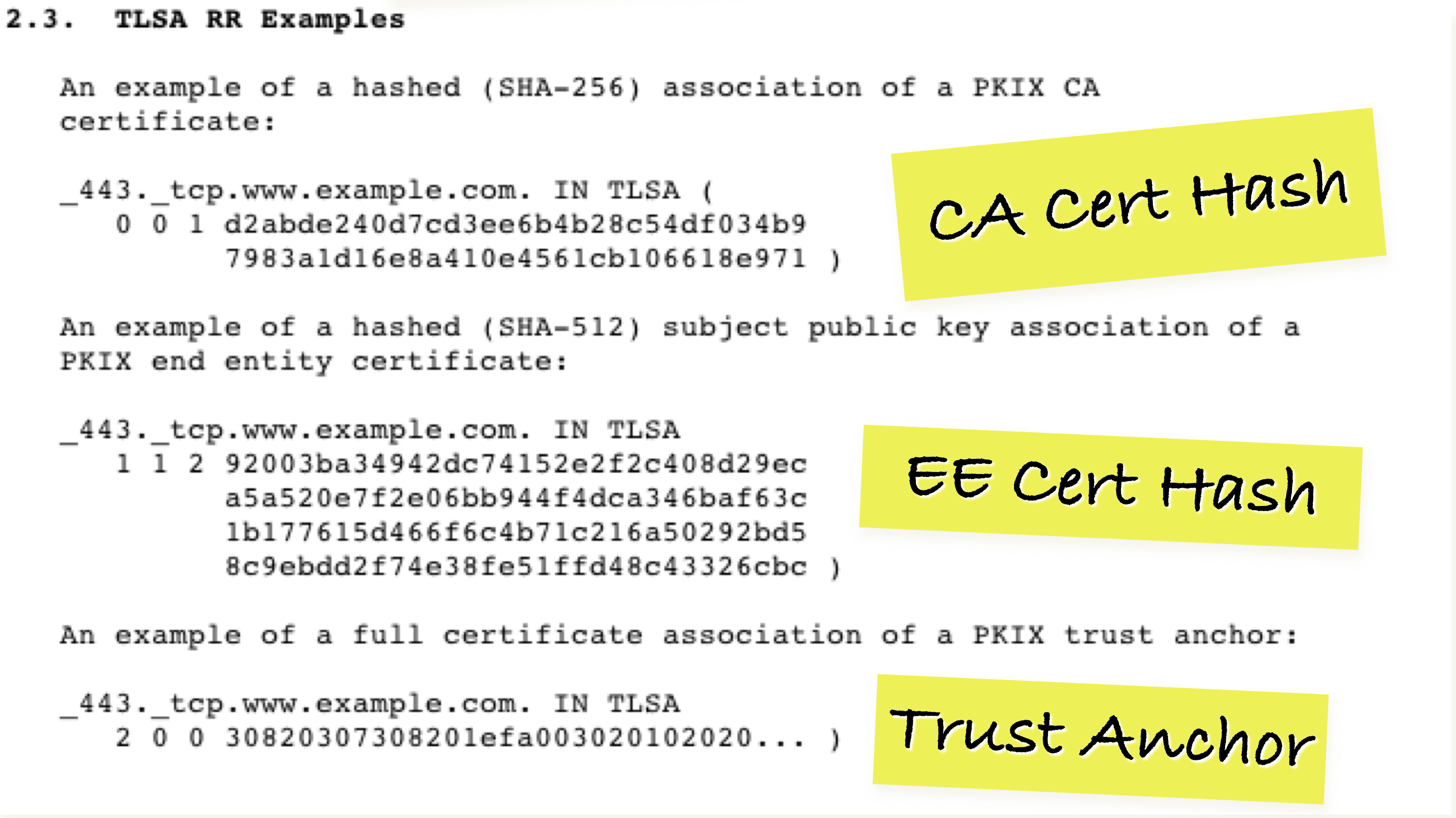


```
[Docs] [txt|pdf] [draft-ietf-dane-ops] [Diff1] [Diff2]

                                                  PROPOSED STANDARD

Internet Engineering Task Force (IETF)                  V. Dukhovni
Request for Comments: 7671                                Two Sigma
Updates: 6698                                           W. Hardaker
Category: Standards Track                                  Pa
ISSN: 2070-1721


    The DNS-Based Authentication of Na                  rotocol:
                    Updates and

Abstract

                        ies and updates the DNS-Based Authentication of
              es (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 )
```

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 )
```

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

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

CA Cert Hash

EE Cert Hash

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

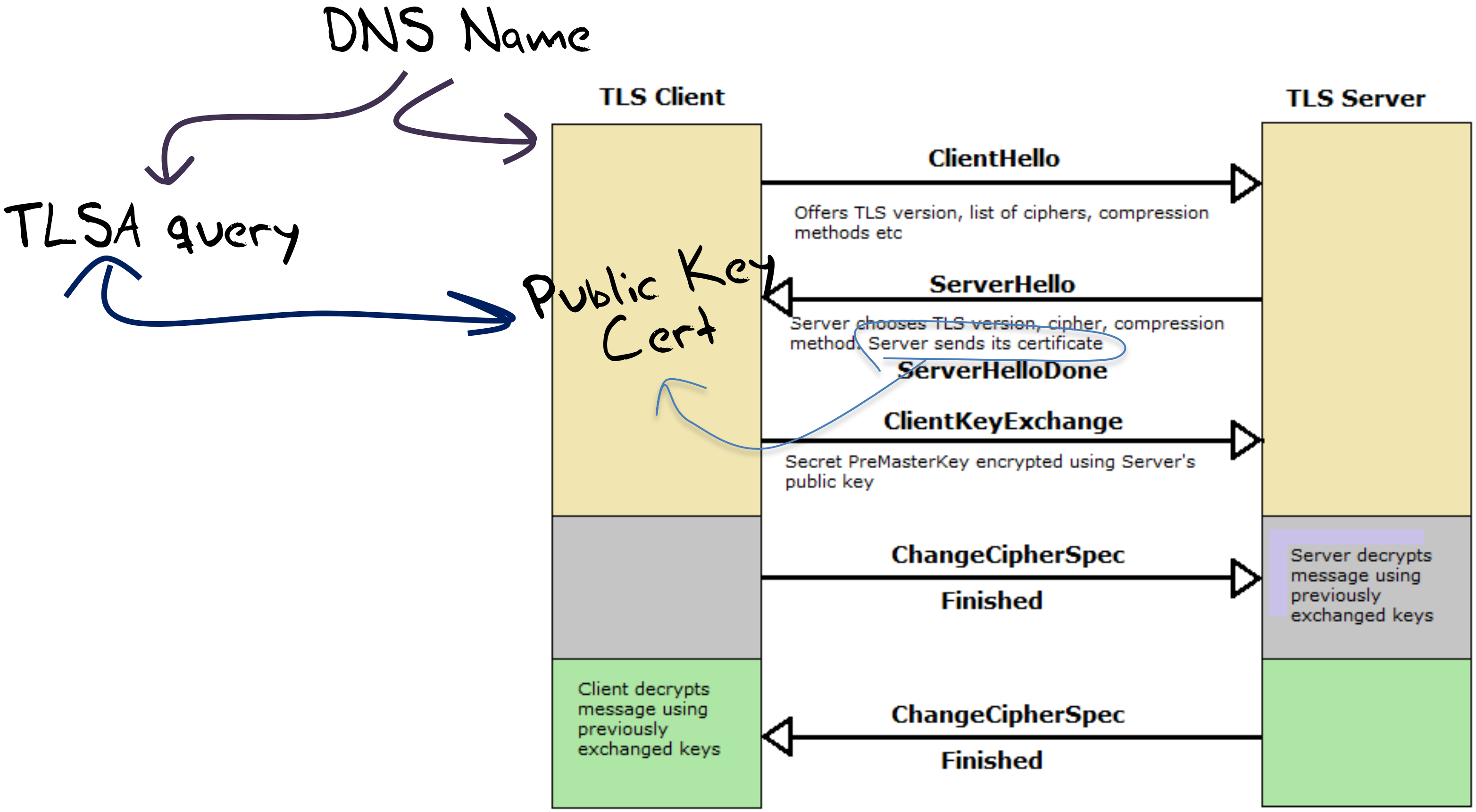
```
; Generate the public key
; Convert it to DER format
; Generate the SHA256 hash
; Add DNS gunk!
```

```
$ /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



TLS Client

TLS Server

ClientHello

Offers TLS version, list of ciphers, compression methods etc

ServerHello

Server chooses TLS version, cipher, compression method. Server sends its certificate

ServerHelloDone

ClientKeyExchange

Secret PreMasterKey encrypted using Server's public key

ChangeCipherSpec

Finished

Server decrypts message using previously exchanged keys

Client decrypts message using previously exchanged keys

ChangeCipherSpec

Finished

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

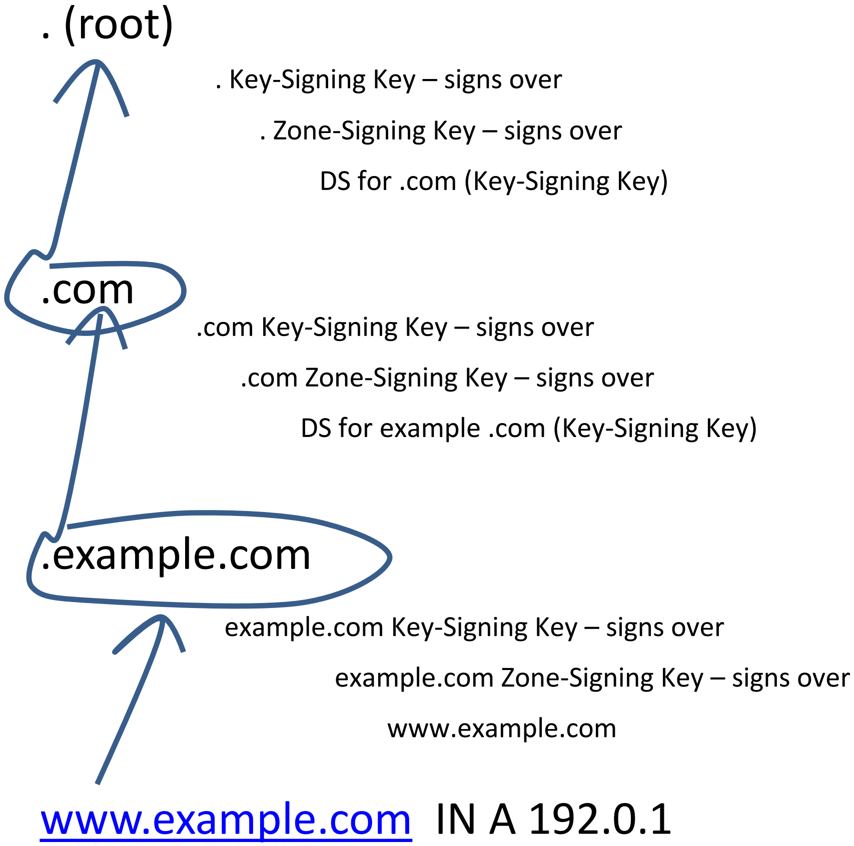
- .com Key-Signing Key – signs over
  - .com Zone-Signing Key – signs over
    - DS for example .com (Key-Signing Key)

.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



. (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

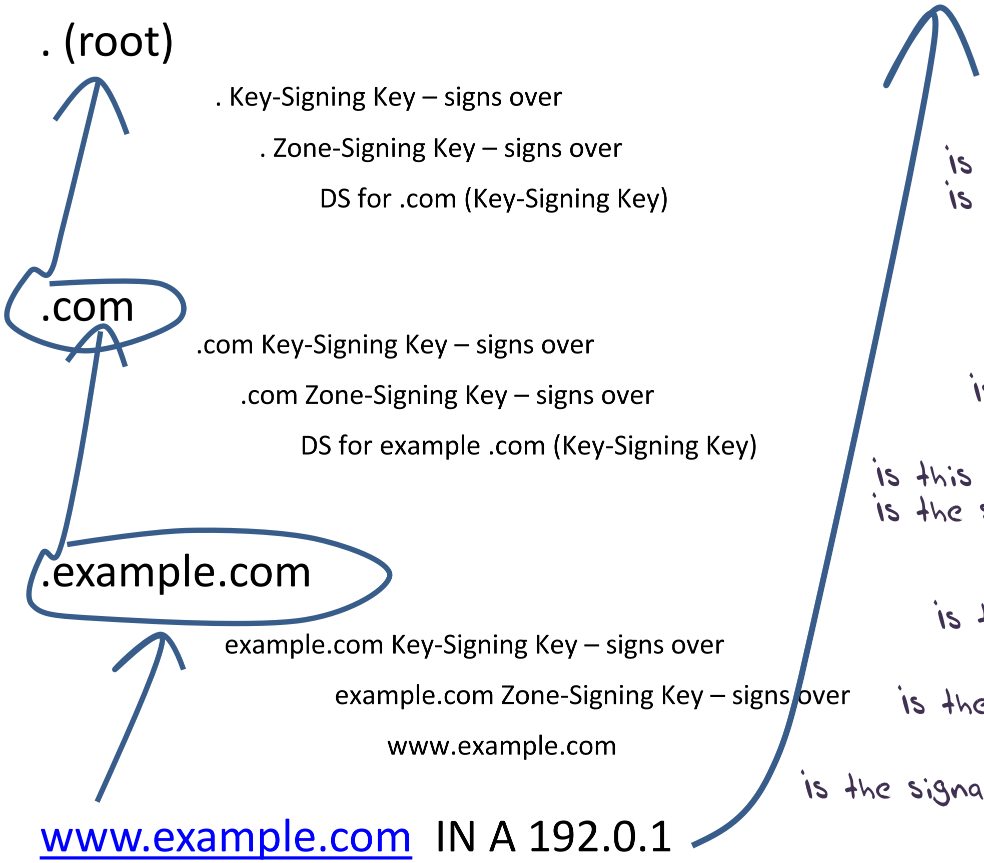
example.com Key-Signing Key – signs over

example.com Zone-Signing Key – signs over

www.example.com

www.example.com IN A 192.0.1

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 IN A 192.0.1

is the KSK for . valid?

is the ZSK for . valid?

is this DS equal to the hash of the KSK?
is the signature for this record valid?

is the KSK for .com valid?

is the ZSK for .com valid?

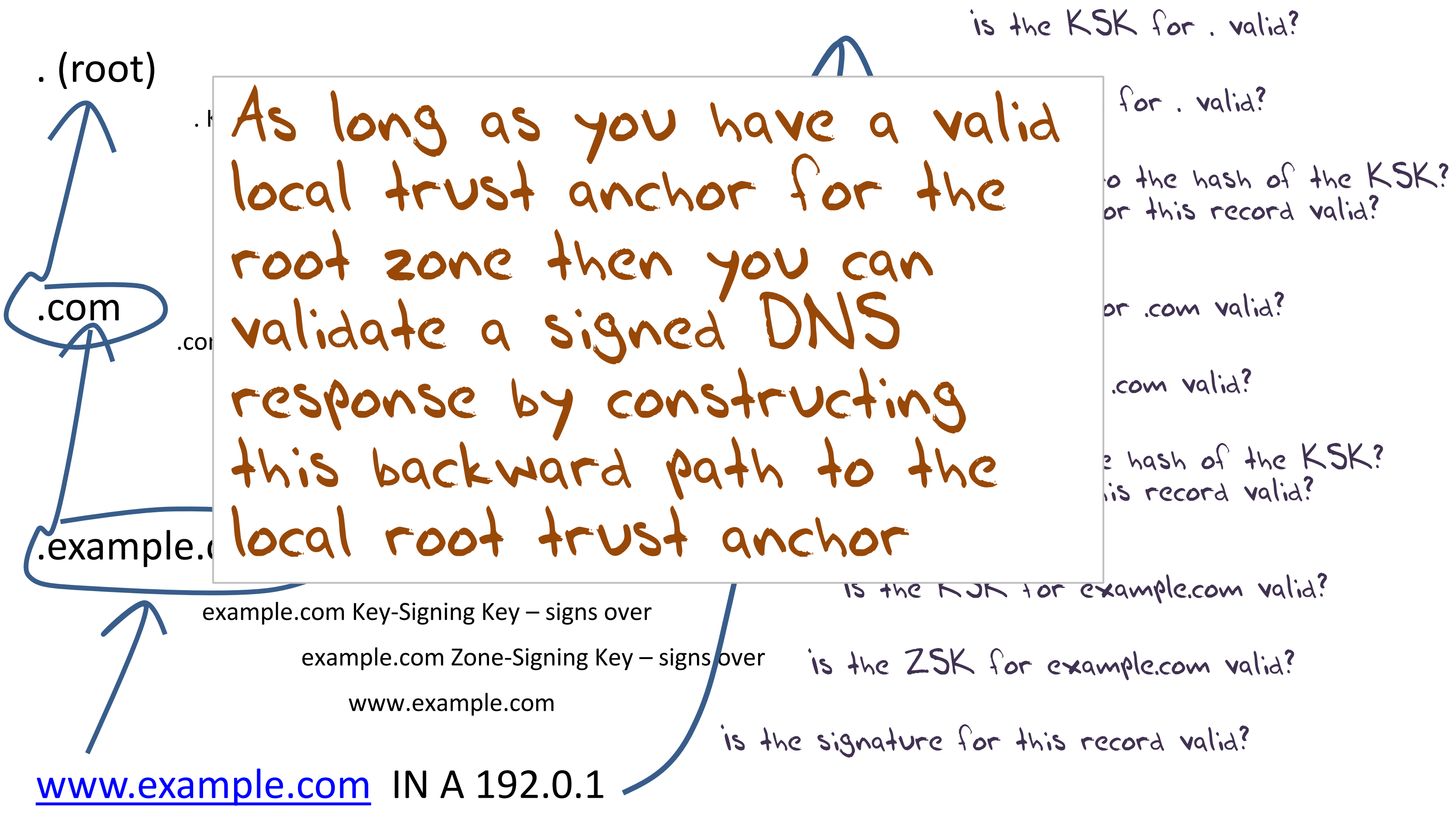
is this DS equal to the hash of the KSK?
is the signature for this record valid?

is the KSK for example.com valid?

is the ZSK for example.com valid?

is the signature for this record valid?

DNSSEC Interlocking Signatures



. (root)

. K

.com

.co

.example.

example.com Key-Signing Key – signs over

example.com Zone-Signing Key – signs over

www.example.com

www.example.com IN A 192.0.1

is the KSK for . valid?

for . valid?

o the hash of the KSK?

or this record valid?

or .com valid?

.com valid?

e hash of the KSK?

is record valid?

is the KSK for example.com valid?

is the ZSK for example.com valid?

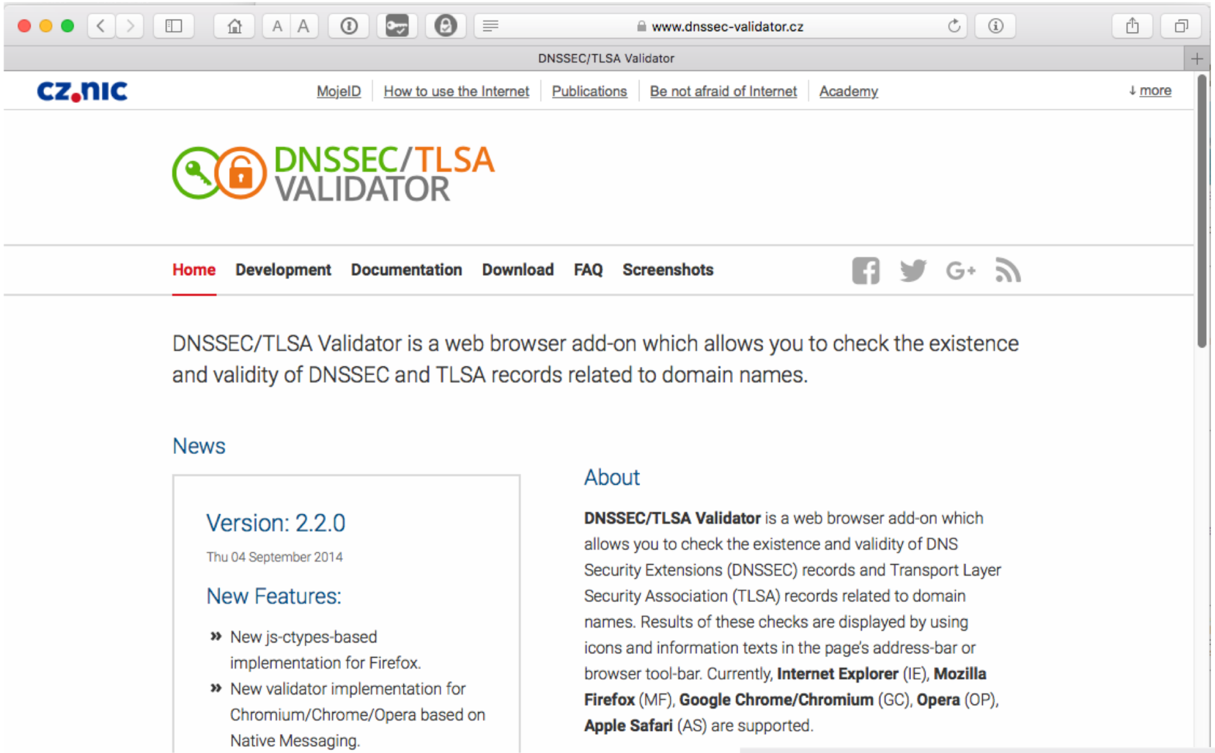
is the signature for this record valid?

As long as you have a valid local trust anchor for the root zone then you can validate a signed DNS response by constructing this backward path to the local root trust anchor

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



DNSSEC/TLSA Validator is a web browser add-on which allows you to check the existence and validity of DNSSEC and TLSA records related to domain names.

News

Version: 2.2.0

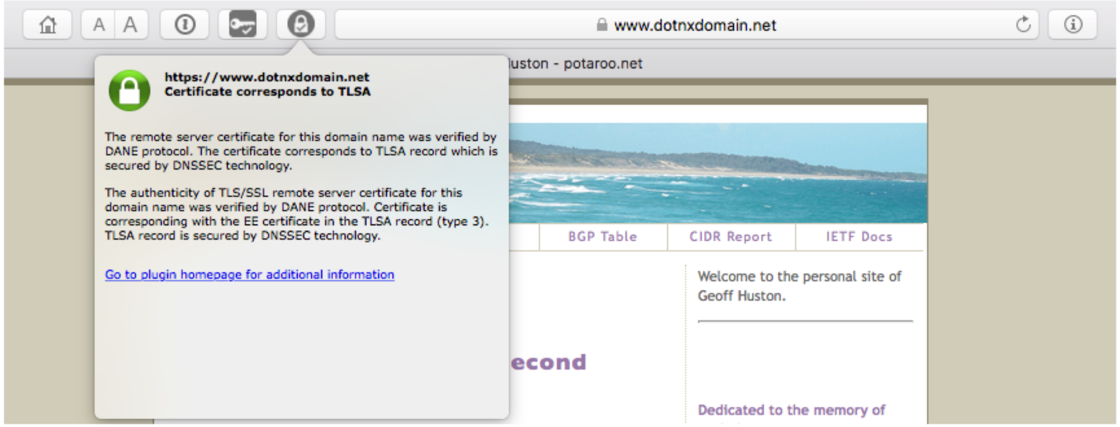
Thu 04 September 2014

New Features:

- New js-ctypes-based implementation for Firefox.
- New validator implementation for Chromium/Chrome/Opera based on Native Messaging.

About

DNSSEC/TLSA Validator is a web browser add-on which allows you to check the existence and validity of DNS Security Extensions (DNSSEC) records and Transport Layer Security Association (TLSA) records related to domain names. Results of these checks are displayed by using icons and information texts in the page's address-bar or browser tool-bar. Currently, **Internet Explorer** (IE), **Mozilla Firefox** (MF), **Google Chrome/Chromium** (GC), **Opera** (OP), **Apple Safari** (AS) are supported.



https://www.dotnxdomain.net
Certificate corresponds to TLSA

The remote server certificate for this domain name was verified by DANE protocol. The certificate corresponds to TLSA record which is secured by DNSSEC technology.

The authenticity of TLS/SSL remote server certificate for this domain name was verified by DANE protocol. Certificate is corresponding with the EE certificate in the TLSA record (type 3). TLSA record is secured by DNSSEC technology.

Go to plugin homepage for additional information

Welcome to the personal site of Geoff Huston.

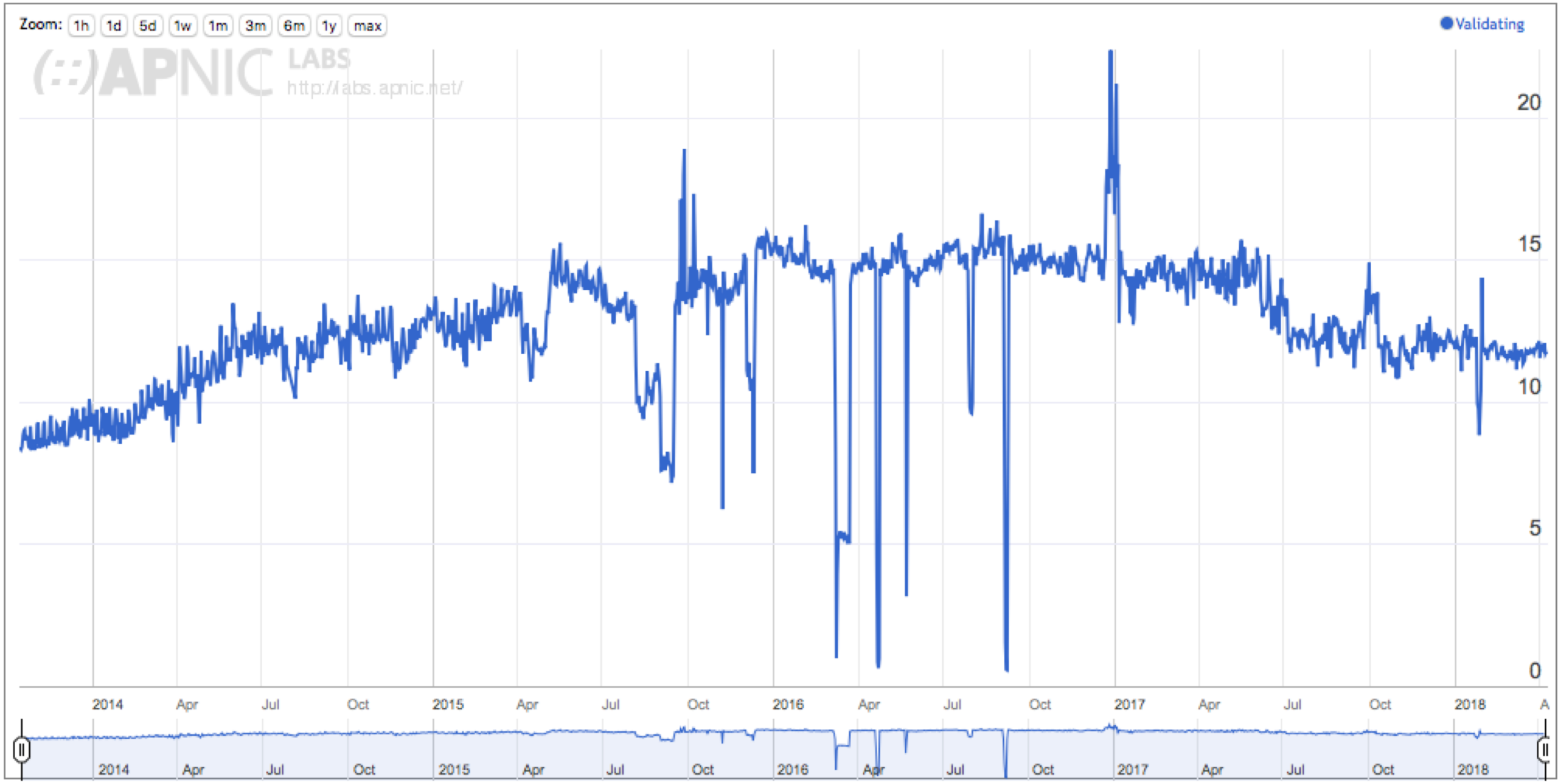
Dedicated to the memory of

So we need DNSSEC as well as DANE…

How much DNSSEC Validation is out there?

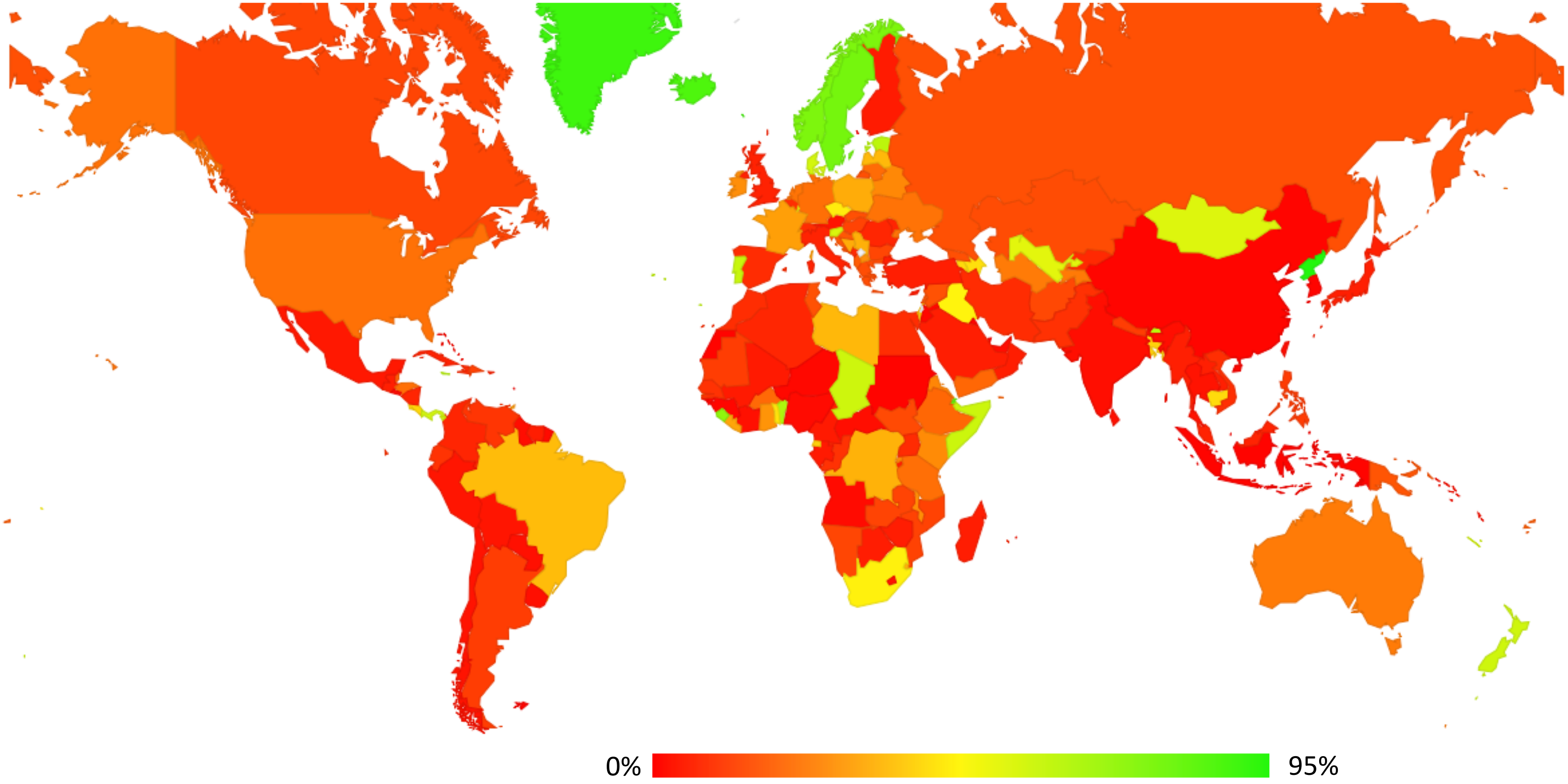
Do we do DNSSEC Validation?

Use of DNSSEC Validation for World (XA)



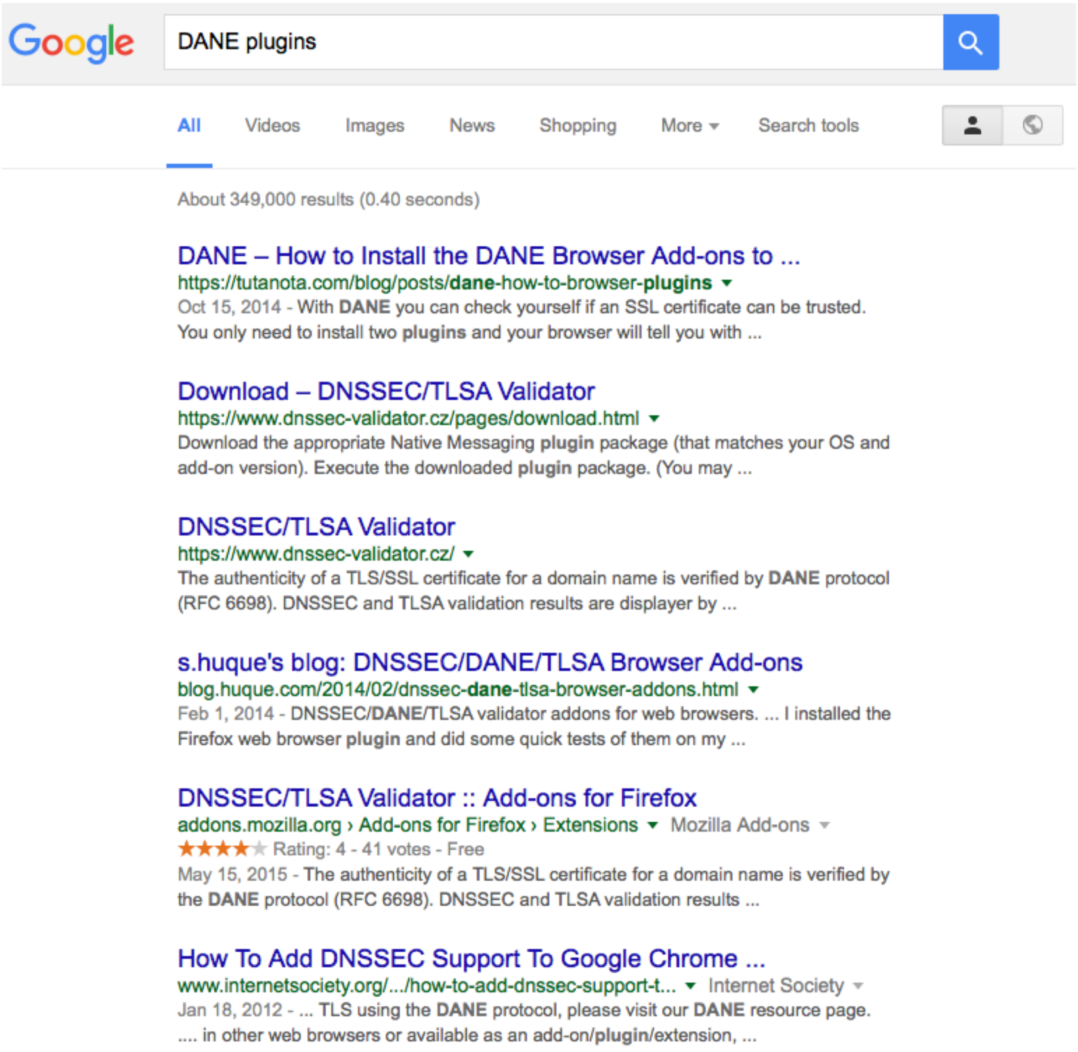
stats.labs.apnic.net/dnssec/XA

Where do we do DNSSEC Validation?



stats.labs.apnic.net/dnssec/XA

Where now?



DANE plugins

About 349,000 results (0.40 seconds)

DANE – How to Install the DANE Browser Add-ons to ...
https://tutanota.com/blog/posts/**dane**-how-to-browser-**plugins**
Oct 15, 2014 - With **DANE** you can check yourself if an SSL certificate can be trusted. You only need to install two **plugins** and your browser will tell you with ...

Download – DNSSEC/TLSA Validator
https://www.dnssec-validator.cz/pages/download.html
Download the appropriate Native Messaging **plugin** package (that matches your OS and add-on version). Execute the downloaded **plugin** package. (You may ...

DNSSEC/TLSA Validator
https://www.dnssec-validator.cz/
The authenticity of a TLS/SSL certificate for a domain name is verified by **DANE** protocol (RFC 6698). DNSSEC and TLSA validation results are displayer by ...

s.huque's blog: DNSSEC/DANE/TLSA Browser Add-ons
blog.huque.com/2014/02/dnssec-**dane**-tlsa-browser-addons.html
Feb 1, 2014 - DNSSEC/**DANE**/TLSA validator addons for web browsers. ... I installed the Firefox web browser **plugin** and did some quick tests of them on my ...

DNSSEC/TLSA Validator :: Add-ons for Firefox
addons.mozilla.org › Add-ons for Firefox › Extensions Mozilla Add-ons
★★★★☆ Rating: 4 - 41 votes - Free
May 15, 2015 - The authenticity of a TLS/SSL certificate for a domain name is verified by the **DANE** protocol (RFC 6698). DNSSEC and TLSA validation results ...

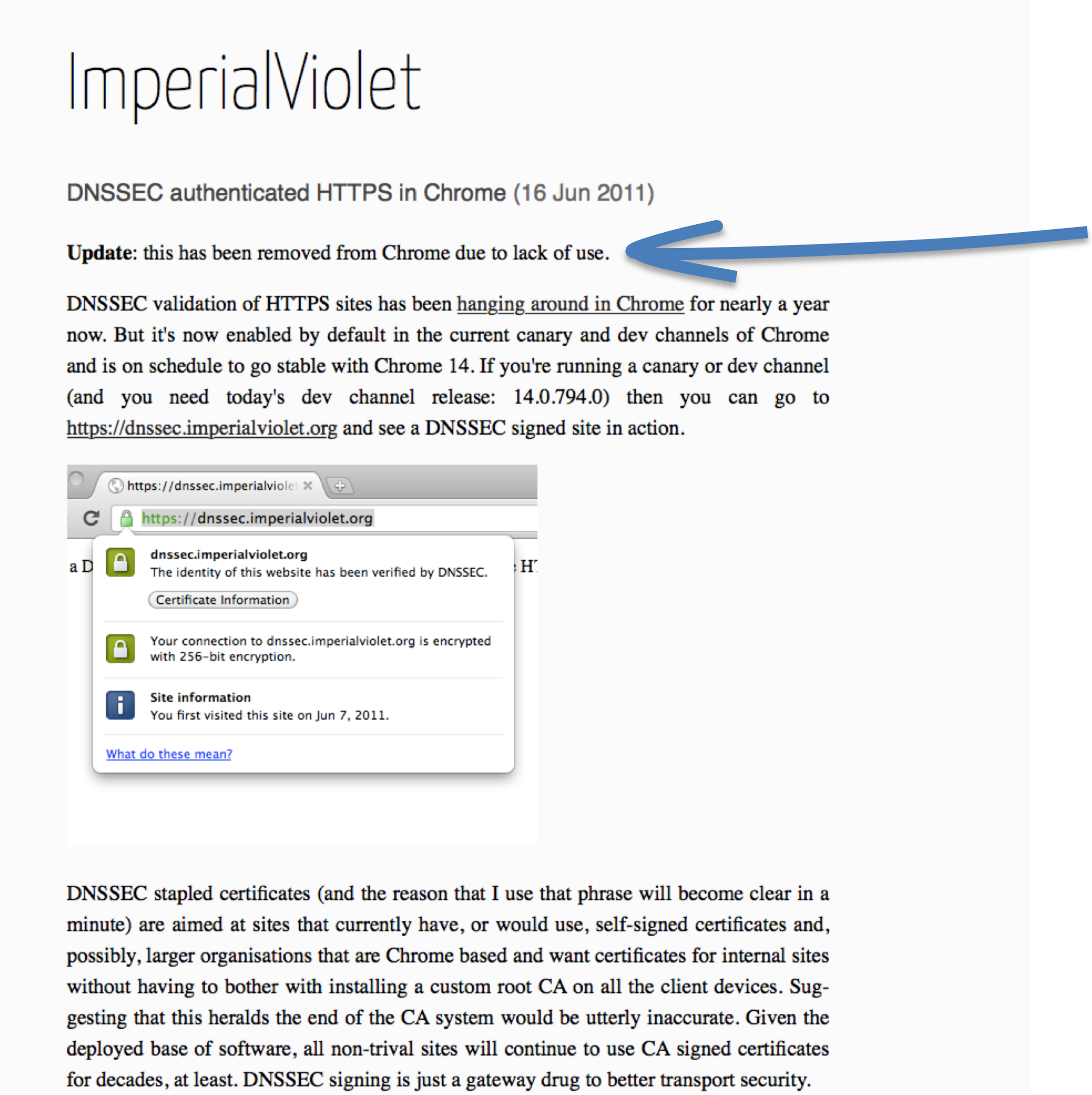
How To Add DNSSEC Support To Google Chrome ...
www.internetsociety.org/.../how-to-add-dnssec-support-t... Internet Society
Jan 18, 2012 - ... TLS using the **DANE** protocol, please visit our **DANE** resource page. .... in other web browsers or available as an add-on/**plugin**/extension, ...

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!

Where now?



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.

https://dnssec.imperialviole

https://dnssec.imperialviolet.org

dnssec.imperialviolet.org
The identity of this website has been verified by DNSSEC.

Certificate Information

Your connection to dnssec.imperialviolet.org is encrypted with 256-bit encryption.

Site information
You first visited this site on Jun 7, 2011.

What do these mean?

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.

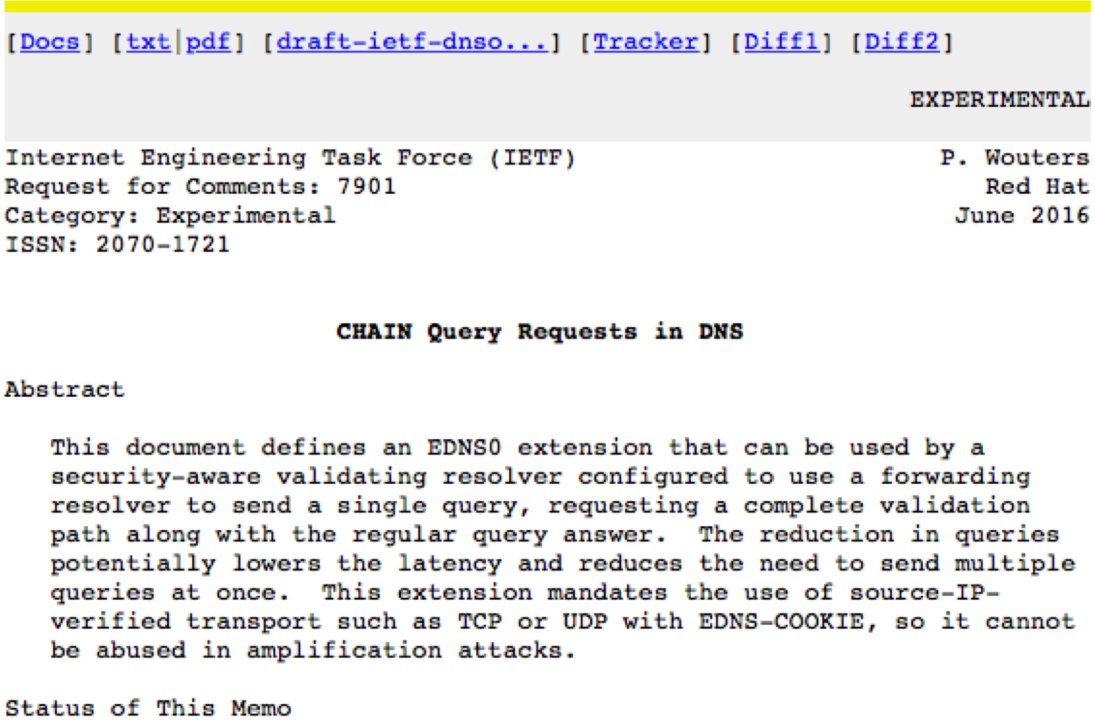
Browser vendors appear to be dragging the chain on DANE support

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…

- We could change the DNS to allow TLS to make efficient use of DANE



[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

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

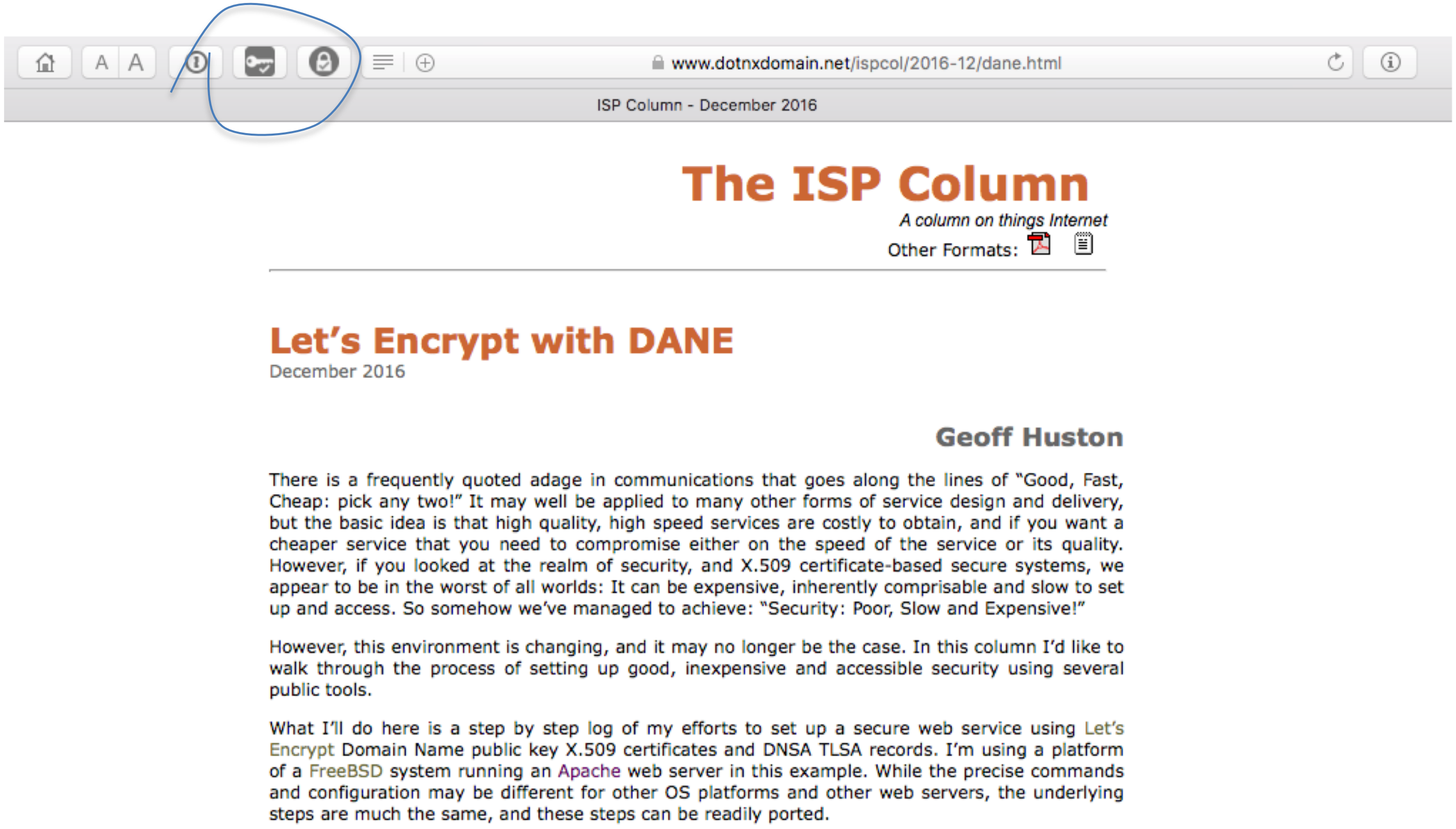
Where now?

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!



www.dotnxdomain.net/ispcol/2016-12/dane.html
ISP Column - December 2016

The ISP Column

A column on things Internet

Other Formats:

Let’s Encrypt with DANE

December 2016

Geoff Huston

There is a frequently quoted adage in communications that goes along the lines of “Good, Fast, Cheap: pick any two!” It may well be applied to many other forms of service design and delivery, but the basic idea is that high quality, high speed services are costly to obtain, and if you want a cheaper service that you need to compromise either on the speed of the service or its quality. However, if you looked at the realm of security, and X.509 certificate-based secure systems, we appear to be in the worst of all worlds: It can be expensive, inherently comprisable and slow to set up and access. So somehow we’ve managed to achieve: “Security: Poor, Slow and Expensive!”

However, this environment is changing, and it may no longer be the case. In this column I’d like to walk through the process of setting up good, inexpensive and accessible security using several public tools.

What I’ll do here is a step by step log of my efforts to set up a secure web service using Let’s Encrypt Domain Name public key X.509 certificates and DNSA TLSA records. I’m using a platform of a FreeBSD system running an Apache web server in this example. While the precise commands and configuration may be different for other OS platforms and other web servers, the underlying steps are much the same, and these steps can be readily ported.

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?