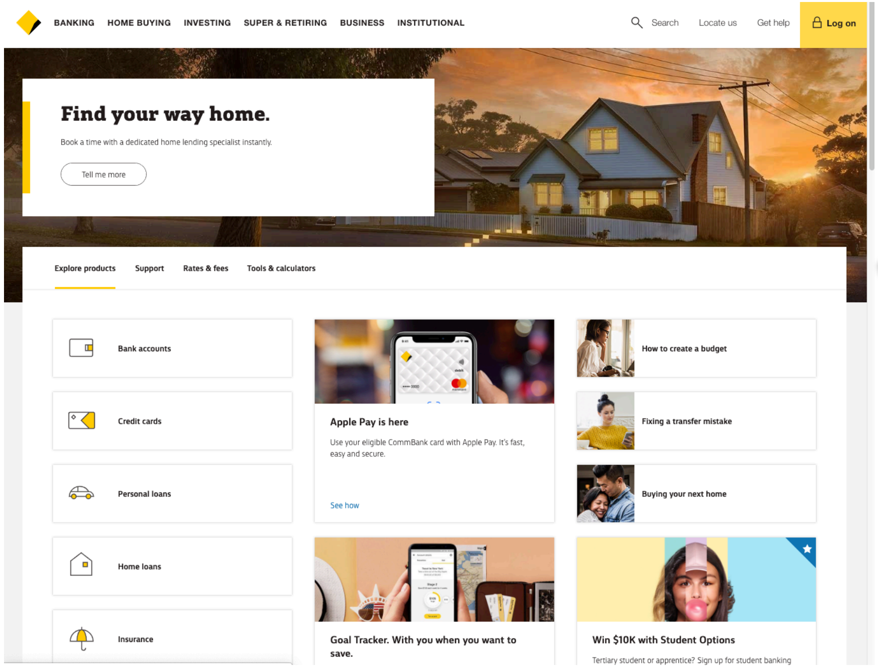
DNSSEC, the DNS and Internet Security

Geoff Huston
Chief Scientist, APNIC
April 2019

Security on the Internet

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



BANKING HOME BUYING INVESTING SUPER & RETIRING BUSINESS INSTITUTIONAL

Search Locate us Get help Log on

Find your way home.

Book a time with a dedicated home lending specialist instantly.

Tell me more

Explore products Support Rates & fees Tools & calculators

Bank accounts

Credit cards

Personal loans

Home loans

Insurance

Apple Pay is here

Use your eligible CommBank card with Apple Pay. It's fast, easy and secure.

See how

Goal Tracker. With you when you want to save.

How to create a budget

Fixing a transfer mistake

Buying your next home

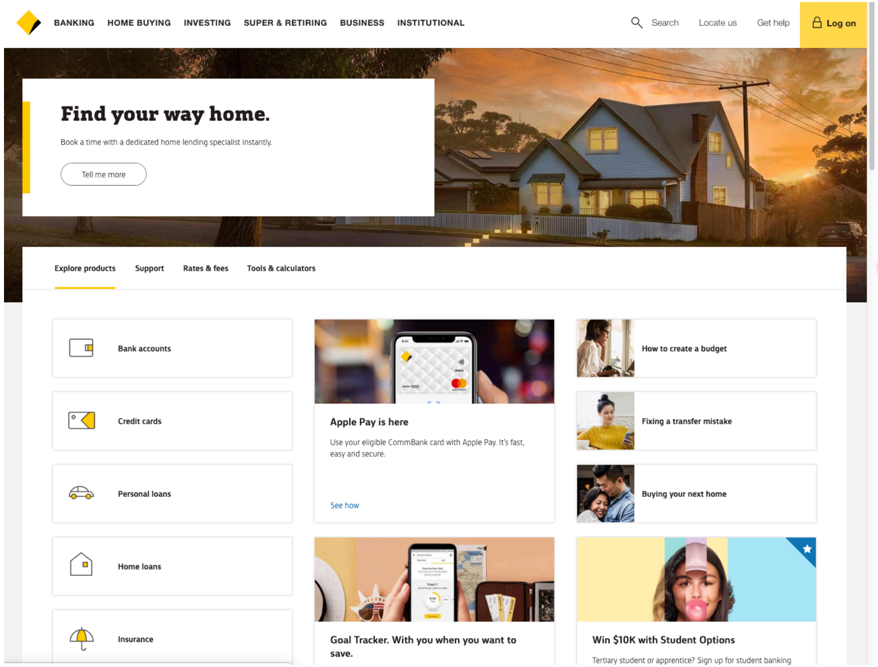
Win $10K with Student Options

Tertiary student or apprentice? Sign up for student banking

My bank

Security on the Internet

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



Find your way home.

Book a time with a dedicated home lending specialist instantly.

Apple Pay is here

Use your eligible CommBank card with Apple Pay. It's fast, easy and secure.

Goal Tracker. With you when you want to save.

Win $10K with Student Options

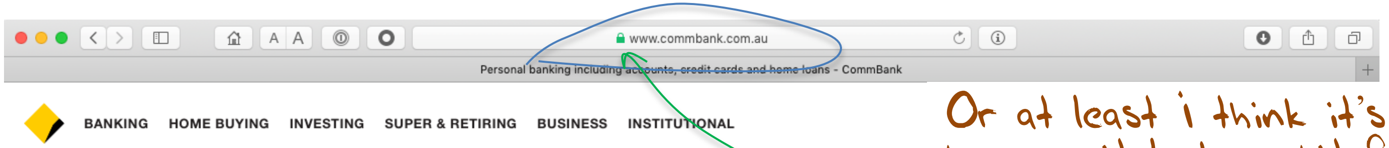
Tertiary student or apprentice? Sign up for student banking

My bank

i hope!

Security on the Internet

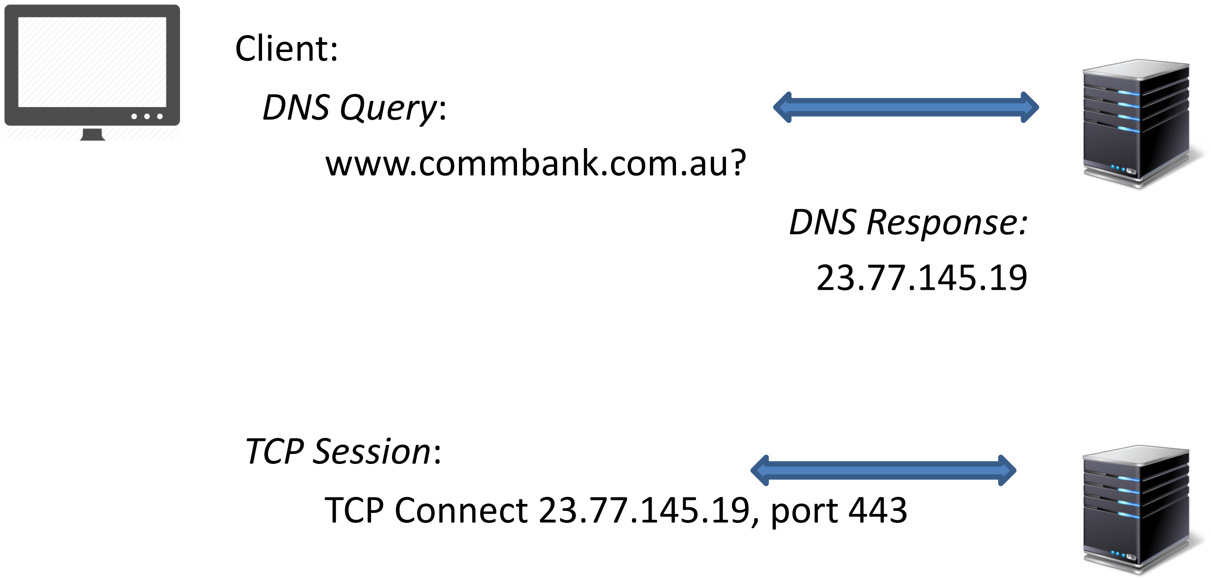
How do you know that you are going to where you thought you were going to?



Or at least i think it's my bank because it looks a bit familiar and there is a totally reassuring green icon of a lock

So it HAS to be my bank – hasn't it?

Connection Steps



Client:

DNS Query:

www.commbank.com.au?

DNS Response:

23.77.145.19

TCP Session:

TCP Connect 23.77.145.19, port 443

Hang on…

```
$ dig -x 23.77.145.19 +short
a23-77-145-19.deploy.static.akamaitechnologies.com.
```

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

Hang on…

```
$ dig -x 23.77.145.19 +short
a23-77-145-19.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 23.77.145.19 is really the “proper” web site for the Commonwealth Bank of Australia and not some dastardly evil scam?

Hang on…

```
$ dig -x 23.77.145.19 +short
a23-77-145-19.deploy.static.akamaitechnologies.com.
```

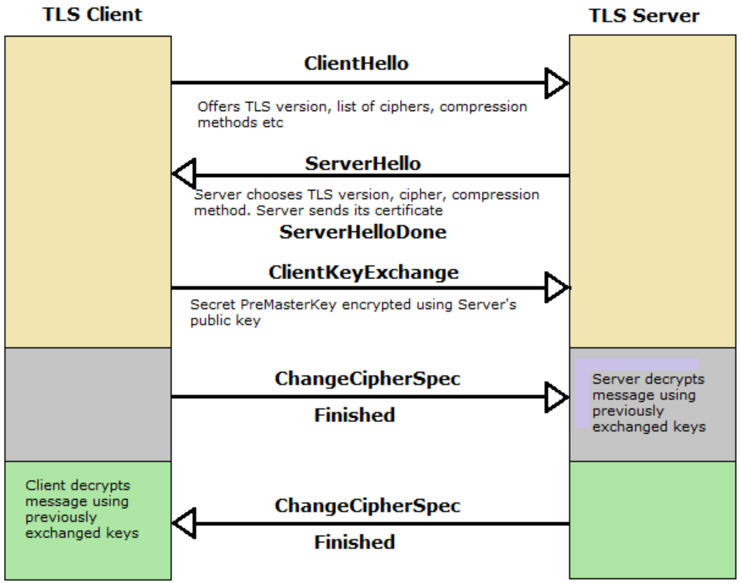
That’s not an IP addresses that was allocated to the Commonwealth Bank!

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 23.77.145.19 is really the “proper” web site for the Commonwealth Bank of Australia and not some dastardly evil scam?

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

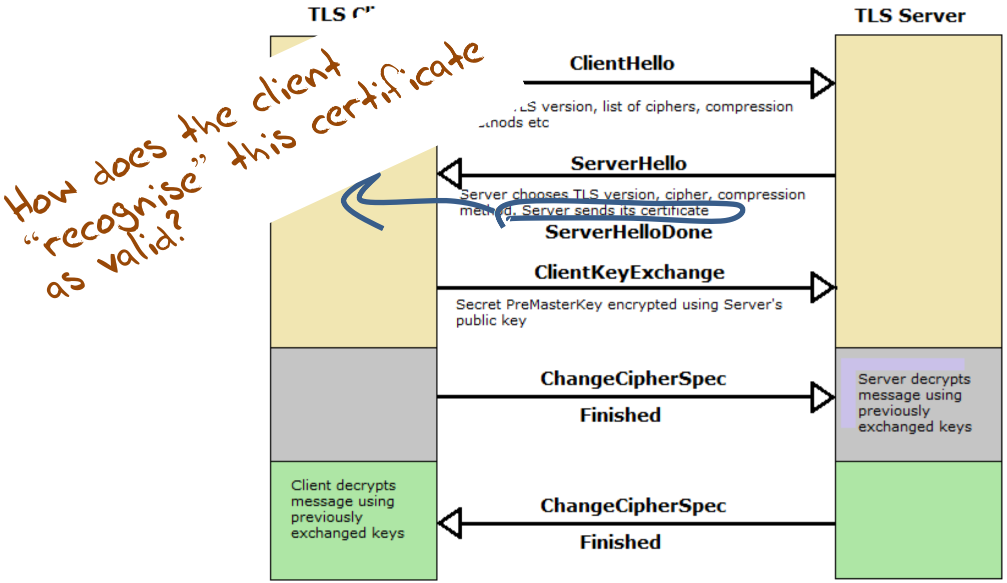
TCP Port 443 Transport Layer Security (TLS) Connections



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/

TCP Port 443 Transport Layer Security (TLS) Connections



TLS Cl

TLS Server

ClientHello

.LS version, list of ciphers, compression .nods 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

ChangeCipherSpec

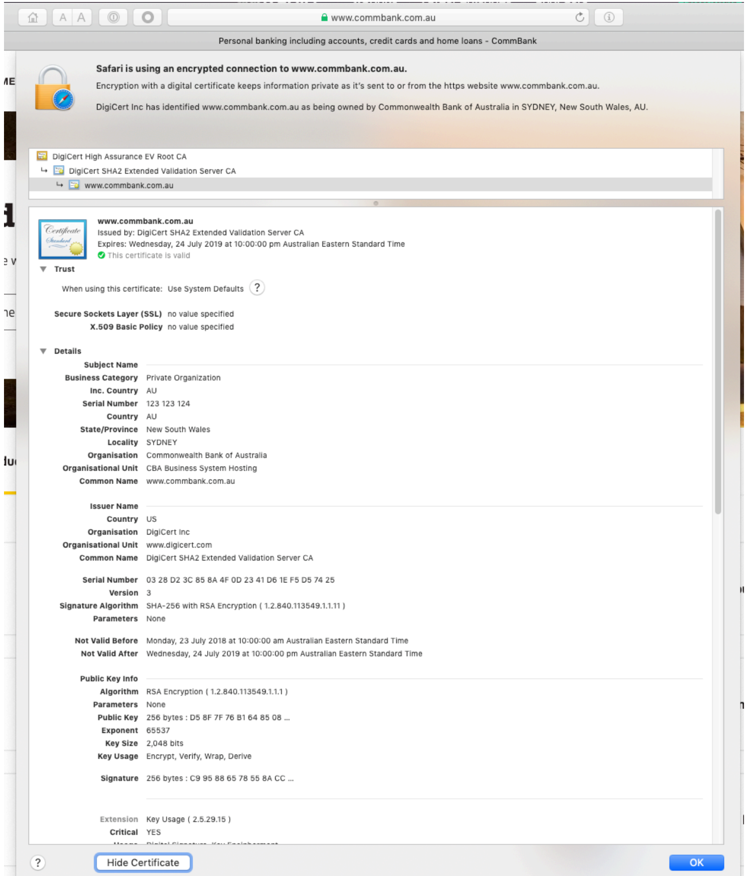
Finished

Client decrypts message using previously exchanged keys

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

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

The Server's Certificate



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

Safari is using an encrypted connection to www.commbank.com.au.

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

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

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

www.commbank.com.au
Issued by: DigiCert SHA2 Extended Validation Server CA
Expires: Wednesday, 24 July 2019 at 10:00:00 pm Australian Eastern Standard Time
✓ This certificate is valid

▼ **Trust**

When using this certificate: Use System Defaults ?

Secure Sockets Layer (SSL) no value specified
X.509 Basic Policy no value specified

▼ **Details**

Subject Name
Business Category Private Organization
Inc. Country AU
Serial Number 123 123 124
Country AU
State/Province New South Wales
Locality SYDNEY
Organisation Commonwealth Bank of Australia
Organisational Unit CBA Business System Hosting
Common Name www.commbank.com.au

Issuer Name
Country US
Organisation DigiCert Inc
Organisational Unit www.digicert.com
Common Name DigiCert SHA2 Extended Validation Server CA

Serial Number 03 28 D2 3C 85 8A 4F 0D 23 41 D6 1E F5 D5 74 25
Version 3
Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters None

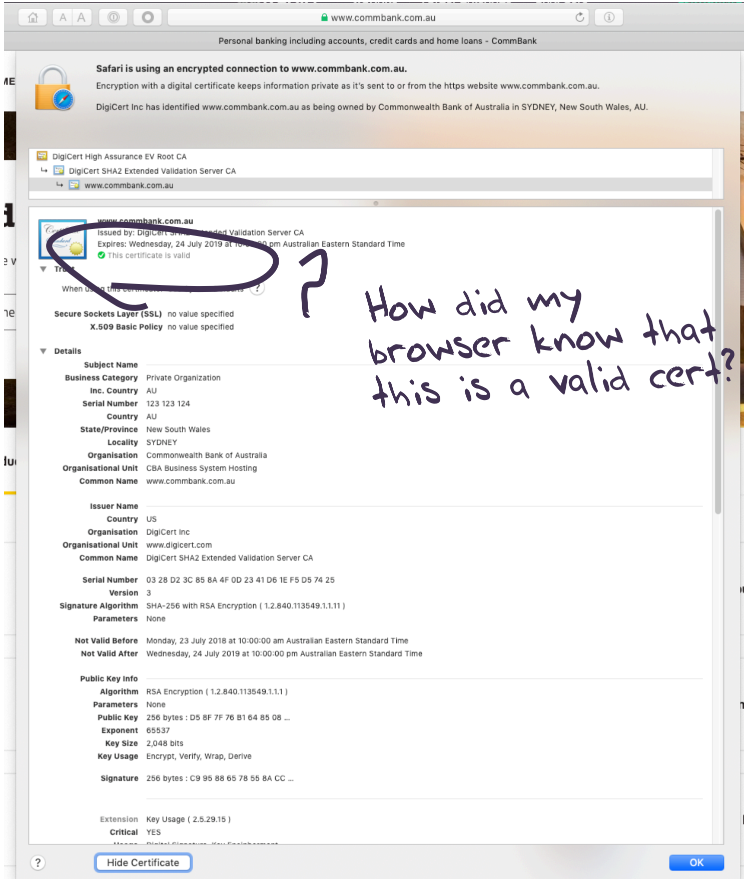
Not Valid Before Monday, 23 July 2018 at 10:00:00 am Australian Eastern Standard Time
Not Valid After Wednesday, 24 July 2019 at 10:00:00 pm Australian Eastern Standard Time

Public Key Info
Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Parameters None
Public Key 256 bytes : D5 8F 7F 76 B1 64 85 08 ...
Exponent 65537
Key Size 2,048 bits
Key Usage Encrypt, Verify, Wrap, Derive

Signature 256 bytes : C9 95 88 65 78 55 8A CC ...

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

The Server's Certificate



www.commbank.com.au
Personal banking including accounts, credit cards and home loans - CommBank

Safari is using an encrypted connection to www.commbank.com.au.

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

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

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

www.commbank.com.au
Issued by: DigiCert SHA2 Extended Validation Server CA
Expires: Wednesday, 24 July 2019 at 10:00:00 pm Australian Eastern Standard Time
✓ This certificate is valid

▼ **Trust**

When using this certificate: Use System Defaults ?

Secure Sockets Layer (SSL) no value specified
X.509 Basic Policy no value specified

▼ **Details**

Subject Name
Business Category Private Organization
Inc. Country AU
Serial Number 123 123 124
Country AU
State/Province New South Wales
Locality SYDNEY
Organisation Commonwealth Bank of Australia
Organisational Unit CBA Business System Hosting
Common Name www.commbank.com.au

Issuer Name
Country US
Organisation DigiCert Inc
Organisational Unit www.digicert.com
Common Name DigiCert SHA2 Extended Validation Server CA

Serial Number 03 28 D2 3C 85 8A 4F 0D 23 41 D6 1E F5 D5 74 25
Version 3
Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters None

Not Valid Before Monday, 23 July 2018 at 10:00:00 am Australian Eastern Standard Time
Not Valid After Wednesday, 24 July 2019 at 10:00:00 pm Australian Eastern Standard Time

Public Key Info
Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Parameters None
Public Key 256 bytes : D5 8F 7F 76 B1 64 85 08 ...
Exponent 65537
Key Size 2,048 bits
Key Usage Encrypt, Verify, Wrap, Derive

Signature 256 bytes : C9 95 88 65 78 55 8A CC ...

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

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

Domain Name Certification

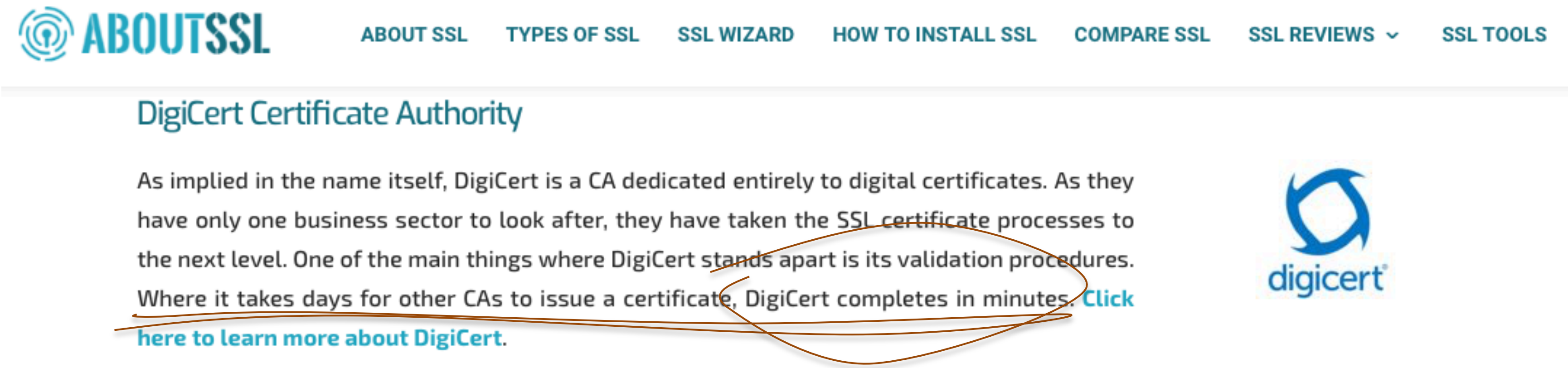
- The Commonwealth Bank of Australia has generated a key pair
- And they passed a Certificate Signing Request to a company called "Digicert" (together with money)
- Digicert is willing to vouch (in a certificate) that the entity who administers the domain name www.commbank.com.au also has a certain public key value (partly because it got paid to do this!)
- So if I can associate this public key with a connection then I have a high degree of confidence that I've connected to the "real" www.commbank.com.au
  - as long as I am also prepared to trust Digicert, and their certificate issuance processes, and that the certificates that they issue are always genuine

Domain Name Certification

- The Commonwealth Bank of Australia has generated a key pair
- And they passed a Certificate Signing Request to a company called "Digicert" (together with money)
- Digicert is willing to vouch (in a certificate) that the entity who administers the domain name www.commbank.com.au also has a certain public key value (partly because it got paid to do this!)
- So if I can associate this public key with a connection then I have a high degree of confidence that I've connected to the "real" www.commbank.com.au
  - as long as I am also prepared to trust Digicert, and their certificate issuance processes, and that the certificates that they issue are always genuine

Why should i trust them?

Digicert

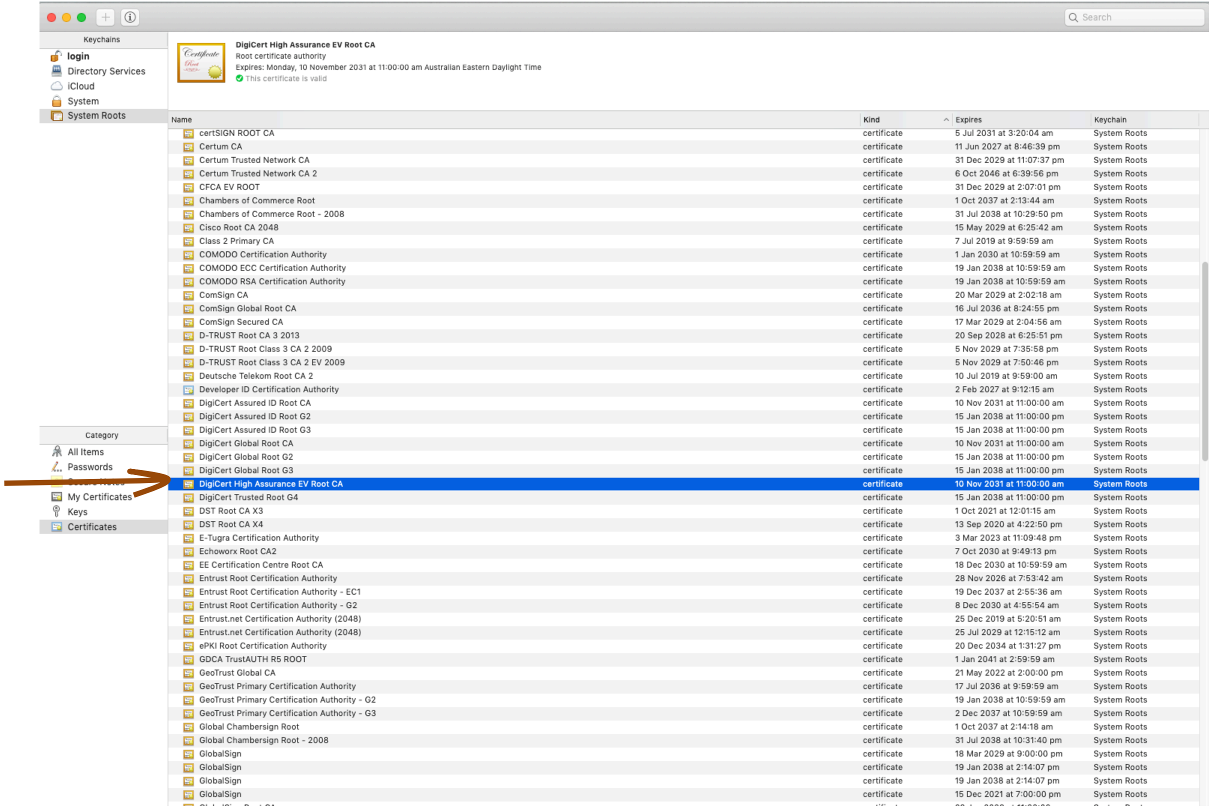


DigiCert Certificate Authority

As implied in the name itself, DigiCert is a CA dedicated entirely to digital certificates. As they have only one business sector to look after, they have taken the SSL certificate processes to the next level. One of the main things where DigiCert stands apart is its validation procedures. Where it takes days for other CAs to issue a certificate, DigiCert completes in minutes. Click here to learn more about DigiCert.

is this the sign of a conscientious CA?

Local Trust



DigiCert High Assurance EV Root CA
Root certificate authority
Expires: Monday, 10 November 2031 at 11:00:00 am Australian Eastern Daylight Time
✓ This certificate is valid

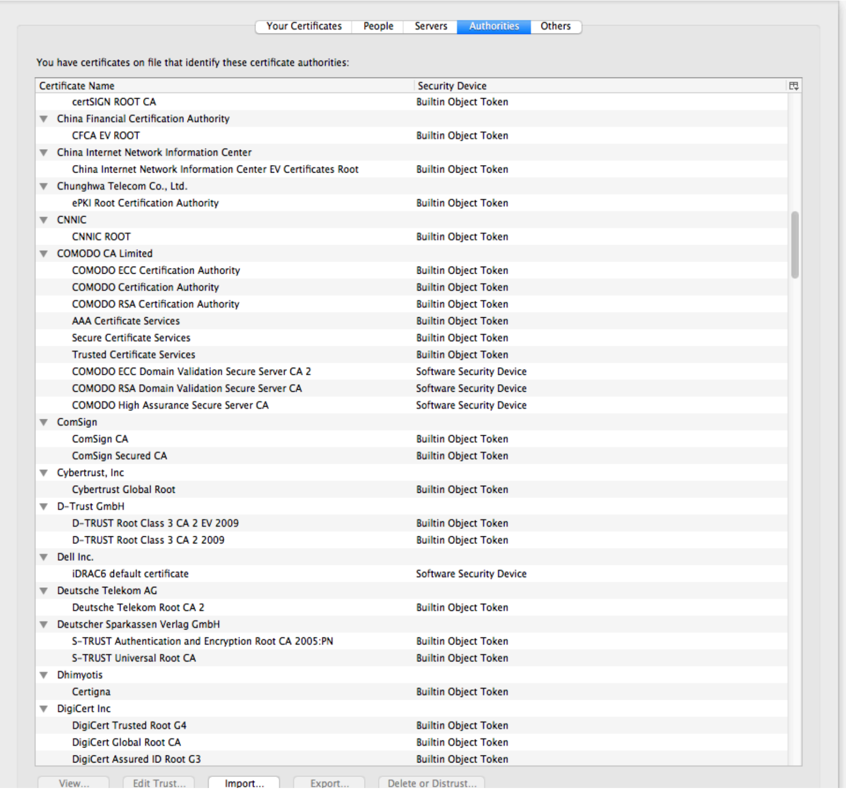
Name	Kind	Expires	Keychain
certSIGN ROOT CA	certificate	5 Jul 2031 at 3:20:04 am	System Roots
Certum CA	certificate	11 Jun 2027 at 8:46:39 pm	System Roots
Certum Trusted Network CA	certificate	31 Dec 2029 at 11:07:37 pm	System Roots
Certum Trusted Network CA 2	certificate	6 Oct 2046 at 6:39:56 pm	System Roots
CFCA EV ROOT	certificate	31 Dec 2029 at 2:07:01 pm	System Roots
Chambers of Commerce Root	certificate	1 Oct 2037 at 2:13:44 am	System Roots
Chambers of Commerce Root - 2008	certificate	31 Jul 2038 at 10:29:50 pm	System Roots
Cisco Root CA 2048	certificate	15 May 2029 at 6:25:42 am	System Roots
Class 2 Primary CA	certificate	7 Jul 2019 at 9:59:59 am	System Roots
COMODO Certification Authority	certificate	1 Jan 2030 at 10:59:59 am	System Roots
COMODO ECC Certification Authority	certificate	19 Jan 2038 at 10:59:59 am	System Roots
COMODO RSA Certification Authority	certificate	19 Jan 2038 at 10:59:59 am	System Roots
ComSign CA	certificate	20 Mar 2029 at 2:02:18 am	System Roots
ComSign Global Root CA	certificate	16 Jul 2036 at 8:24:55 pm	System Roots
ComSign Secured CA	certificate	17 Mar 2029 at 2:04:56 am	System Roots
D-TRUST Root CA 3 2013	certificate	20 Sep 2028 at 6:25:51 pm	System Roots
D-TRUST Root Class 3 CA 2 2009	certificate	5 Nov 2029 at 7:35:58 pm	System Roots
D-TRUST Root Class 3 CA 2 EV 2009	certificate	5 Nov 2029 at 7:50:46 pm	System Roots
Deutsche Telekom Root CA 2	certificate	10 Jul 2019 at 9:59:00 am	System Roots
Developer ID Certification Authority	certificate	2 Feb 2027 at 9:12:15 am	System Roots
DigiCert Assured ID Root CA	certificate	10 Nov 2031 at 11:00:00 am	System Roots
DigiCert Assured ID Root G2	certificate	15 Jan 2038 at 11:00:00 pm	System Roots
DigiCert Assured ID Root G3	certificate	15 Jan 2038 at 11:00:00 pm	System Roots
DigiCert Global Root CA	certificate	10 Nov 2031 at 11:00:00 am	System Roots
DigiCert Global Root G2	certificate	15 Jan 2038 at 11:00:00 pm	System Roots
DigiCert Global Root G3	certificate	15 Jan 2038 at 11:00:00 pm	System Roots
DigiCert High Assurance EV Root CA	certificate	10 Nov 2031 at 11:00:00 am	System Roots
DigiCert Trusted Root G4	certificate	15 Jan 2038 at 11:00:00 pm	System Roots
DST Root CA X3	certificate	1 Oct 2021 at 12:01:15 am	System Roots
DST Root CA X4	certificate	13 Sep 2020 at 4:22:50 pm	System Roots
E-Tugra Certification Authority	certificate	3 Mar 2023 at 11:09:48 pm	System Roots
Echoworx Root CA2	certificate	7 Oct 2030 at 9:49:13 pm	System Roots
EE Certification Centre Root CA	certificate	18 Dec 2030 at 10:59:59 am	System Roots
Entrust Root Certification Authority	certificate	28 Nov 2026 at 7:53:42 am	System Roots
Entrust Root Certification Authority - EC1	certificate	19 Dec 2037 at 2:55:36 am	System Roots
Entrust Root Certification Authority - G2	certificate	8 Dec 2030 at 4:55:54 am	System Roots
Entrust.net Certification Authority (2048)	certificate	25 Dec 2019 at 5:20:51 am	System Roots
Entrust.net Certification Authority (2048)	certificate	25 Jul 2029 at 12:15:12 am	System Roots
ePKI Root Certification Authority	certificate	20 Dec 2034 at 1:31:27 pm	System Roots
GDCA TrustAUTH R5 ROOT	certificate	1 Jan 2041 at 2:59:59 am	System Roots
GeoTrust Global CA	certificate	21 May 2022 at 2:00:00 pm	System Roots
GeoTrust Primary Certification Authority	certificate	17 Jul 2036 at 9:59:59 am	System Roots
GeoTrust Primary Certification Authority - G2	certificate	19 Jan 2038 at 10:59:59 am	System Roots
GeoTrust Primary Certification Authority - G3	certificate	2 Dec 2037 at 10:59:59 am	System Roots
Global Chambersign Root	certificate	1 Oct 2037 at 2:14:18 am	System Roots
Global Chambersign Root - 2008	certificate	31 Jul 2038 at 10:31:40 pm	System Roots
GlobalSign	certificate	18 Mar 2029 at 9:00:00 pm	System Roots
GlobalSign	certificate	19 Jan 2038 at 2:14:07 pm	System Roots
GlobalSign	certificate	19 Jan 2038 at 2:14:07 pm	System Roots
GlobalSign	certificate	15 Dec 2021 at 7:00:00 pm	System Roots

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

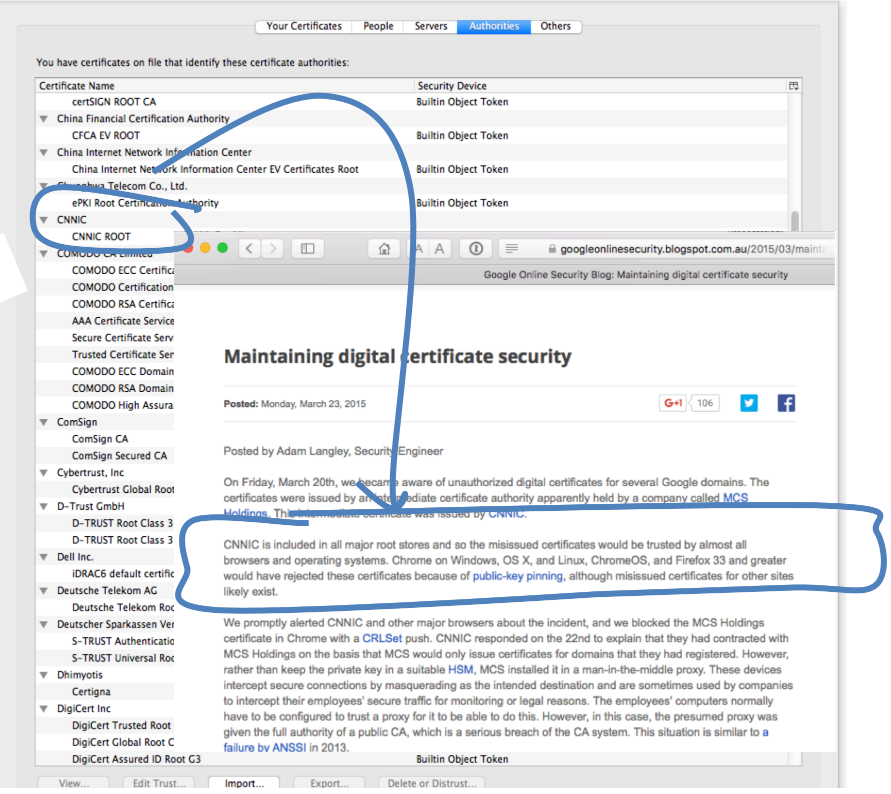
* cre·du·li·ty
/krəˈd(y)o͞olədē/
noun
a tendency to be too ready to believe that something is real or true.

Local Credulity

That’s a big list of people to Trust

Are they all trustable?

Evidently Not!



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	
▼ COMODO CA Limited	
COMODO ECC Certifica	
COMODO Certification	
COMODO RSA Certifica	
AAA Certificate Service	
Secure Certificate Serv	
Trusted Certificate Ser	
COMODO ECC Domain	
COMODO RSA Domain	
COMODO High Assura	
▼ ComSign	
ComSign CA	
ComSign Secured CA	
▼ Cybertrust, Inc	
Cybertrust Global Root	
▼ D-Trust GmbH	
D-TRUST Root Class 3	
D-TRUST Root Class 3	
▼ Dell Inc.	
iDRAC6 default certific	
▼ Deutsche Telekom AG	
Deutsche Telekom Roo	
▼ Deutscher Sparkassen Ver	
S-TRUST Authenticatio	
S-TRUST Universal Roo	
▼ Dhimyotis	
Certigna	
▼ DigiCert Inc	
DigiCert Trusted Root	
DigiCert Global Root C	
DigiCert Assured ID Root G3	Builtin Object Token

googleonlinesecurity.blogspot.com.au/2015/03/mainta

Google Online Security Blog: Maintaining digital certificate security

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.

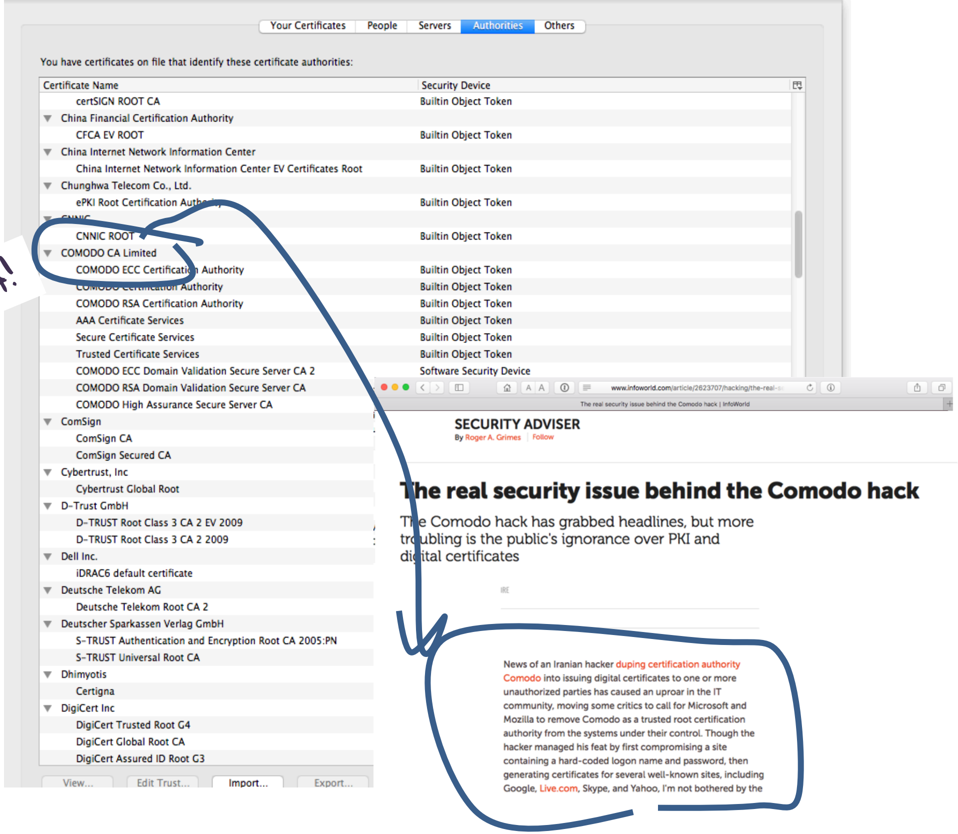
We promptly alerted CNNIC and other major browsers about the incident, and we blocked the MCS Holdings certificate in Chrome with a CRLSet push. CNNIC responded on the 22nd to explain that they had contracted with MCS Holdings on the basis that MCS would only issue certificates for domains that they 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 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	
COMODO High Assurance Secure Server CA	
▼ ComSign	
ComSign CA	
ComSign Secured CA	
▼ Cybertrust, Inc	
Cybertrust Global Root	
▼ D-Trust GmbH	
D-TRUST Root Class 3 CA 2 EV 2009	
D-TRUST Root Class 3 CA 2 2009	
▼ Dell Inc.	
iDRAC6 default certificate	
▼ Deutsche Telekom AG	
Deutsche Telekom Root CA 2	
▼ Deutscher Sparkassen Verlag GmbH	
S-TRUST Authentication and Encryption Root CA 2005:PN	
S-TRUST Universal Root CA	
▼ Dhimyotis	
Certigna	
▼ DigiCert Inc	
DigiCert Trusted Root G4	
DigiCert Global Root CA	
DigiCert Assured ID Root G3	

View...
Edit Trust...
Import...
Export...
www.infoworld.com/article/2623707/hacking/the-real-s
The real security issue behind the Comodo hack | InfoWorld

SECURITY ADVISER

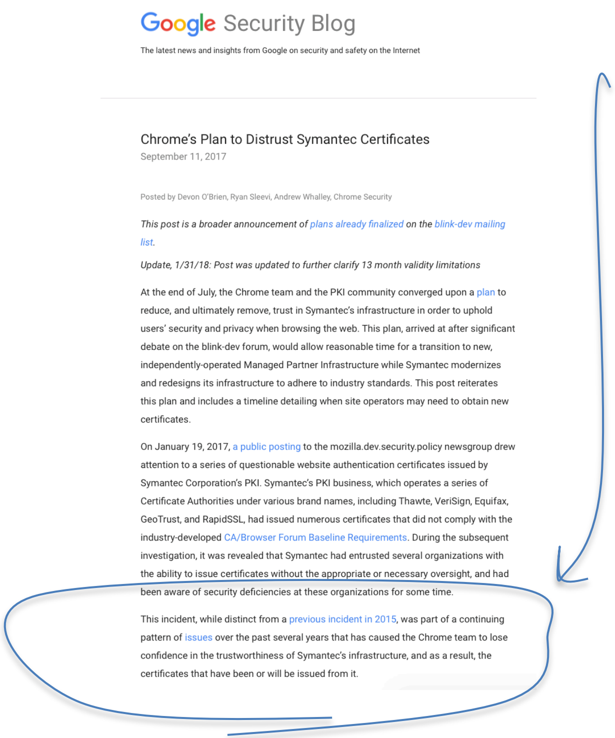
By Roger A. Grimes | Follow

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

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

Credulity



Google Security Blog

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

Chrome's Plan to Distrust Symantec Certificates

September 11, 2017

Posted by Devon O'Brien, Ryan Sleevi, Andrew Whalley, Chrome Security

This post is a broader announcement of plans already finalized on the blink-dev mailing list.

Update, 1/31/18: Post was updated to further clarify 13 month validity limitations

At the end of July, the Chrome team and the PKI community converged upon a plan to reduce, and ultimately remove, trust in Symantec's infrastructure in order to uphold users' security and privacy when browsing the web. This plan, arrived at after significant debate on the blink-dev forum, would allow reasonable time for a transition to new, independently-operated Managed Partner Infrastructure while Symantec modernizes and redesigns its infrastructure to adhere to industry standards. This post reiterates this plan and includes a timeline detailing when site operators may need to obtain new certificates.

On January 19, 2017, a public posting to the mozilla.dev.security.policy newsgroup drew attention to a series of questionable website authentication certificates issued by Symantec Corporation's PKI. Symantec's PKI business, which operates a series of Certificate Authorities under various brand names, including Thawte, VeriSign, Equifax, GeoTrust, and RapidSSL, had issued numerous certificates that did not comply with the industry-developed CA/Browser Forum Baseline Requirements. During the subsequent investigation, it was revealed that Symantec had entrusted several organizations with the ability to issue certificates without the appropriate or necessary oversight, and had been aware of security deficiencies at these organizations for some time.

This incident, while distinct from a previous incident in 2015, was part of a continuing pattern of issues over the past several years that has caused the Chrome team to lose confidence in the trustworthiness of Symantec's infrastructure, and as a result, the certificates that have been or will be issued from it.

So i don't really have a say at all as to what i trust

For my Chrome browser "the Google team" makes that decision on my behalf

For my Mac "the Apple team" determine what i trust

For my Windows platform i trust what Microsoft trusts

Are you feeling better about all this now?

With unpleasant consequences when it all goes wrong

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
- That means that your browser may 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
- That means that your browser may 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 the digital certificate
- That means that your browser may allow ANY CA to be used to 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!

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

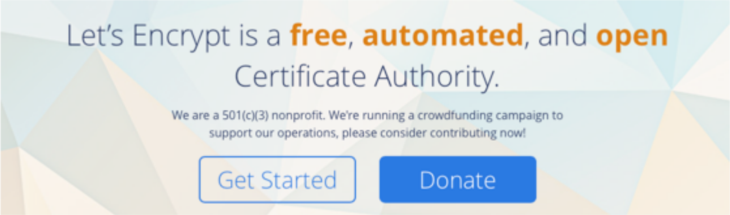
Where CA’s compete with each other for market share

And quality offers no protection

Than what ‘wins’ in the market?

Sustainable
Resilient
Secure
Privacy
Trusted
Cheap!

Cheap Won!

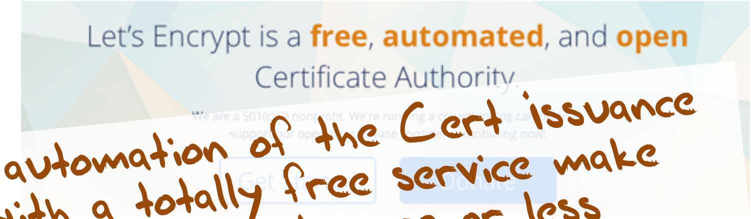


Let's Encrypt is a **free**, **automated**, and **open** Certificate Authority.

We are a 501(c)(3) nonprofit. We're running a crowdfunding campaign to support our operations, please consider contributing now!

www.letsencrypt.org

Cheap Won!



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

We are a 501(c)(3) nonprofit. We're running a crowdfunding campaign to support our operations. Please consider contributing now!

www.letsencrypt.org

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

Well, we now know the answer!

What's the problem

- If ANY CA can issue a valid certificate for ANY Domain Name then the system is compromised:
  - No matter who I choose to be my CA, any CA can issue a certificate for my Domain Name
  - The system is only as strong as the weakest link
- So maybe we need to ‘**pin**’ a domain name to a given CA

CA Pinning

Chrome and in-code pinning

HPKP

CAA

Certificate Transparency Logs

Like the iPv6 transition, we have devised numerous approaches to this problem

CA Pinning

Chrome and in-code pinning

Doesn't scale

HPKP

TOFU is useless

CAA

Rogue CAs are not stopped

Certificate Transparency Logs

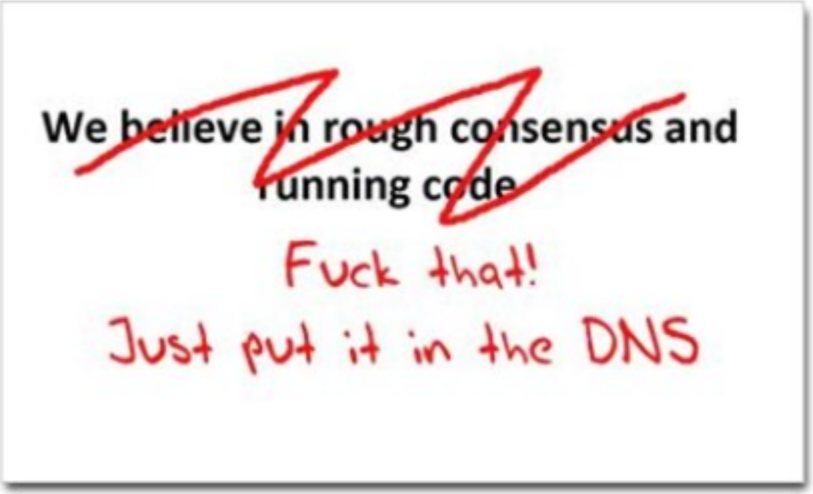
Too little, too slowly

Like the iPv6 transition, we have devised numerous approaches to this problem

But none of them are terribly effective!

Where now?

Use the DNS?



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

Fuck that!

Just put it in the DNS

cafepress.com/nxdomain

Seriously … just use the DNS Luke!*

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

* Ok, that was a cheap shot – my apologies to the Star Wars franchise!

Seriously

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

- Why not query the DNS for the issuer CA?

or

- Why not query the DNS for the hash of the domain name cert?

or

- Why not query the DNS for the hash of the domain name subject public key info?

Seriously

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

– Why not query the DNS for the iss

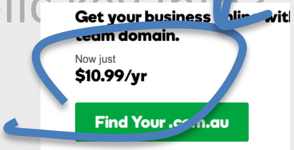
or

– Why not q s for the hash of the domain name cert?

or

– Why not query the DNS for the hash e subject public key info?

Who needs CA's anyway?

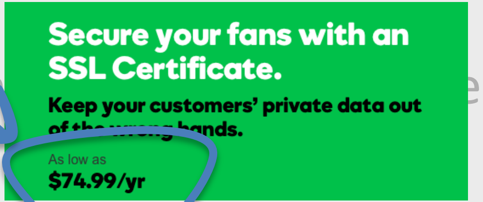


Get your business online with team domain.

Now just

$10.99/yr

Find Your .com.au



Secure your fans with an SSL Certificate.

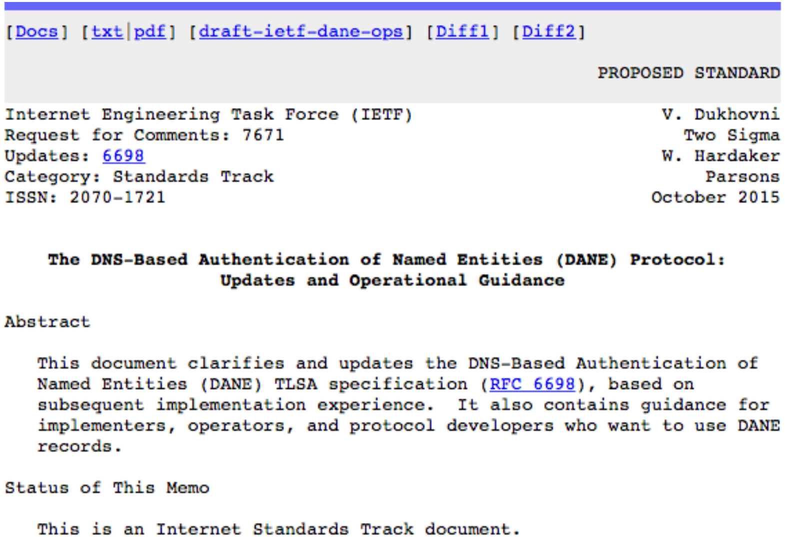
Keep your customers' private data out of the wrong hands.

As low as

$74.99/yr

DANE

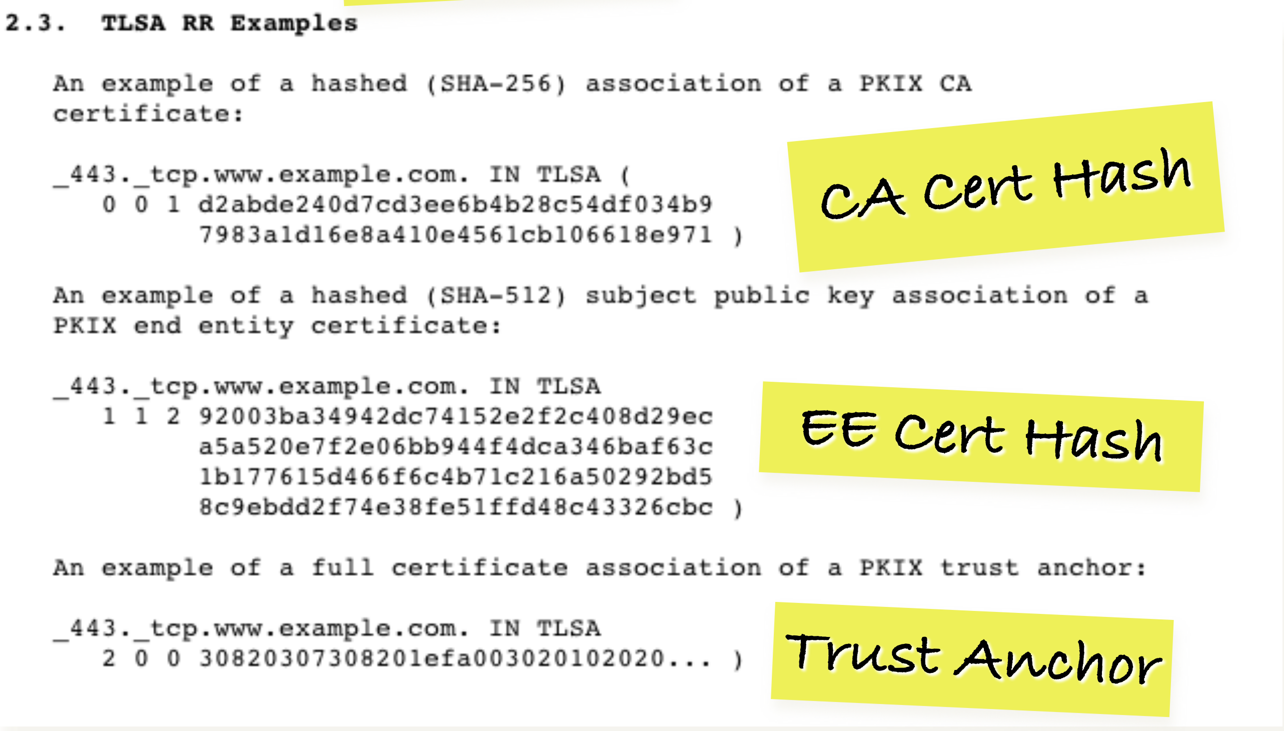
- Using the DNS to associated domain name public key certificates with domain name



```
[Docs] [txt|pdf] [draft-ietf-dane-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                                      Parsons
ISSN: 2070-1721                                           October 2015


    The DNS-Based Authentication of Named Entities (DANE) Protocol:
                    Updates and Operational Guidance

Abstract

   This document clarifies and updates the DNS-Based Authentication of
   Named Entities (DANE) TLSA specification (RFC 6698), based on
   subsequent 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.
```

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:

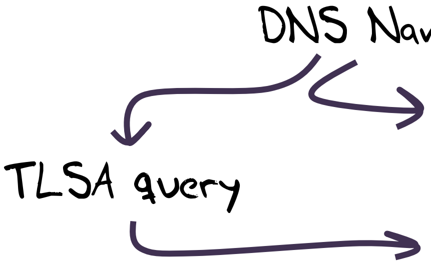
```
_443._tcp.www.example.com. IN TLSA
   2 0 0 30820307308201efa003020102020... )
```

Trust Anchor

TLS with DANE

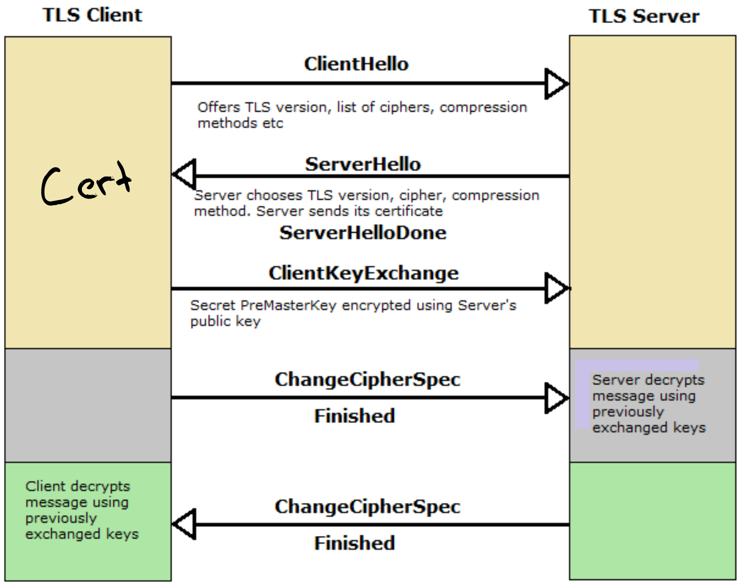
- Client receives server cert in Server Hello
  - *Client lookups the DNS for the TLSA Resource Record of the domain name*
  - *Client validates the presented certificate against the TLSA RR*
- Client performs Client Key exchange

TLS Connections



DNSSEC/TLSA Validator 2.2.0.2.1-signed
by CZ.NIC Labs

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


TLS Client

TLS Server

Cert

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

ChangeCipherSpec

Finished

Client decrypts message using previously exchanged keys

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

Just one problem…

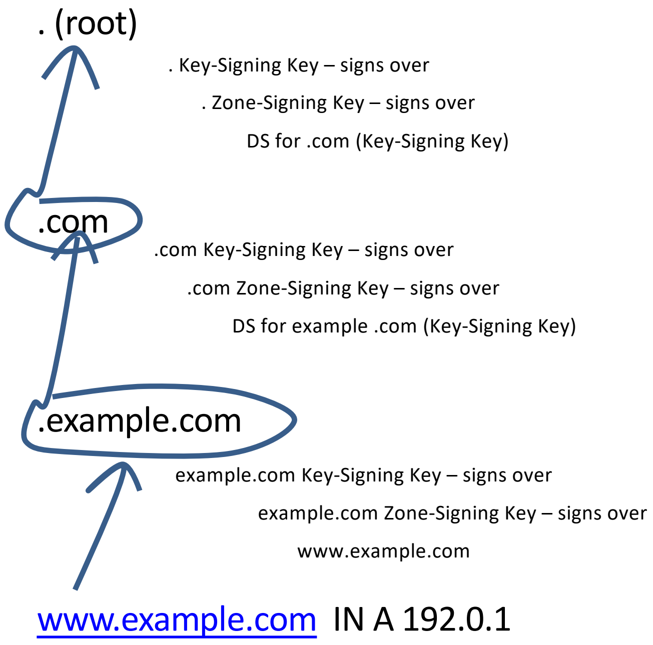
- The DNS is full of liars and lies!
- And this can compromise the integrity of public key information embedded in the DNS
- Unless we fix the DNS we are no better off than before with these TLSA records!

Just one answer…

- 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!

- 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

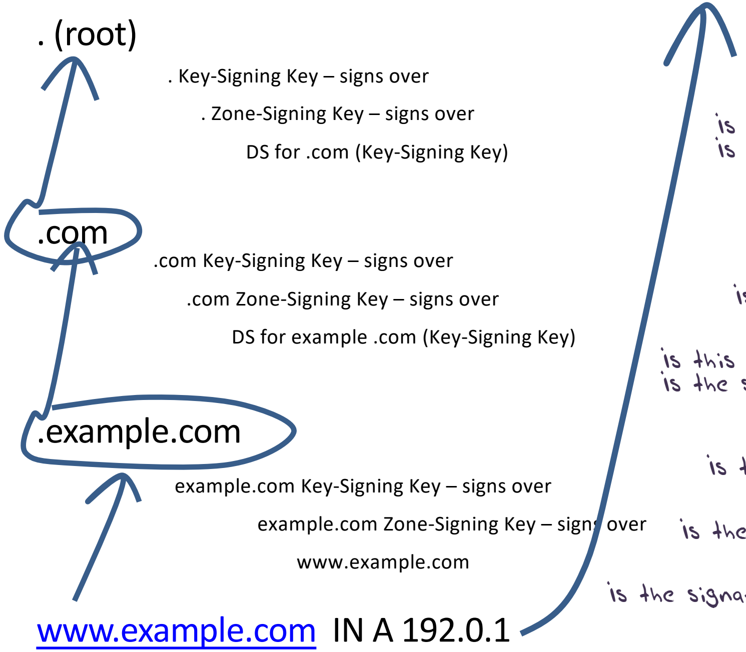
example.com Key-Signing Key – signs over

example.com Zone-Signing Key – signs over

www.example.com

www.example.com 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?

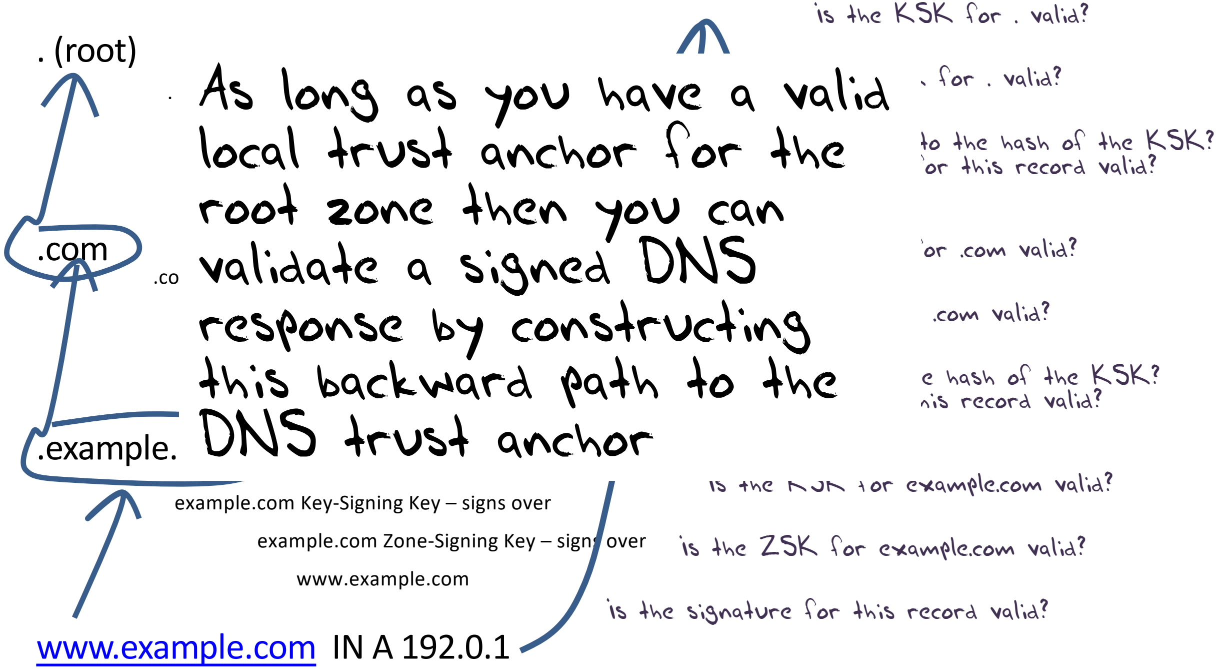
is this DS equal to the hash of the KSK?
is the signature for this record valid?

is the KSK for example.com valid?

is the ZSK for example.com valid?

is the signature for this record valid?

DNSSEC Interlocking Signatures



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 DNS trust anchor

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?

to the hash of the KSK?
'or this record valid?

'or .com valid?

.com valid?

e hash of the KSK?
nis record valid?

is the KSK for example.com valid?

is the ZSK for example.com valid?

is the signature for this record valid?

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

Alternatively - Look! No DNS!

- The Server packages server cert, TLSA record and the DNSSEC credential chain in a single bundle for TLS
- Client receives bundle in TLS 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

Why DNSSEC?

DNSSEC was devised in response to the possibility of cache poisoning attacks on the DNS (the so-called “Kaminsky attack”)

> but the combination of randomized source ports, free Domain name certificates and the use of TLS made that problem go away!

But a reliable and trustable DNS can be very useful for the larger issue of Internet Security

DNSSEC provides us with such a tool for the DNS

Next Steps

- Security for the Internet is an ongoing task
- We know the current WebPKI is hopelessly compromised, and adversaries have been successful in mounting attacks on Internet infrastructure
- The approach of placing Domain Name Keys in a DNSSEC-secured DNS record seems to hold considerable promise to improve the integrity of Domain Name Keys
  - But it’s still a work-in-progress, not a completed solution

Some Practical Suggestions

Some things you can do today:

- Use a Name registrar that at a minimum uses multi-factor authentication and Registry Lock
- Sign your DNS name with DNSSEC
- Obtain Domain Name certificates
- Use TLS and DKIM in all your services
- Turn on DNSSEC Validation in your DNS resolvers

Some Practical Suggestions

Some things you can do today:

- Use a Name registrar that at a minimum uses multi-factor authentication and Registry Lock

Because if I can take over your name registration then I can create the potential to assume control over your online services

So your name registration credentials needs to be more than a simple password and an email address if the name is important to you and your users

Some Practical Suggestions

Some things you can do today:

- Use a Name registrar that at a minimum uses multi-factor authentication and Registry Lock
- Sign your DNS name with DNSSEC

I can now place information in the DNS that clients can trust as being my information

Some Practical Suggestions

Some things you can do today:

- Use a Name registrar that at a minimum uses multi-factor authentication and Registry Lock
- Sign your DNS name with DNSSEC
- Obtain Domain Name certificates

Lets Encrypt is effective - use it!

Some Practical Suggestions

Some things you can do today:

- Use a Name registrar that at a minimum uses multi-factor authentication and Registry Lock
- Sign your DNS name with DNSSEC
- Obtain Domain Name certificates
- Use TLS and DKIM in all your services

Passing data over the Internet in the clear is so
Irresponsible these days!

Some Practical Suggestions

Some things you can do today:

- Use a Name registrar that at a minimum uses multi-factor authentication and Registry Lock
- Sign your DNS name with DNSSEC
- Obtain Domain Name certificates
- Use TLS and DKIM in all your services
- Turn on DNSSEC Validation in your DNS resolvers

Don’t accept signed DNS responses that cannot be validated

Some Practical Suggestions

Some things you can do today:

- Use a Name registrar that at a minimum uses multi-factor authentication and Registry Lock
- Sign your DNS name with DNSSEC
- Obtain Domain Name certificates
- Use TLS and DKIM in all your services
- Turn on DNSSEC Validation in your DNS resolvers

That's it!