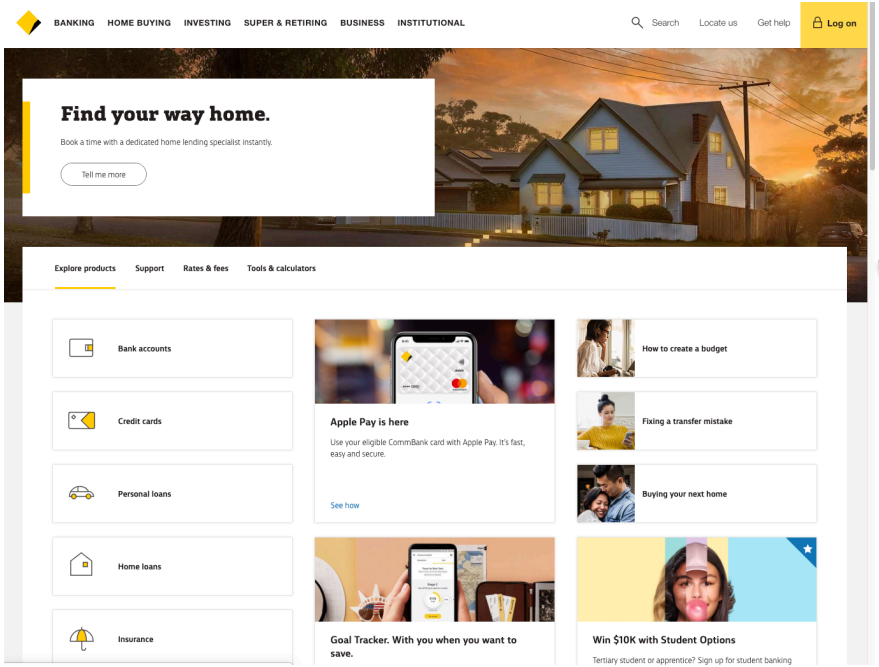
DNSSEC, the DNS and Internet Security

Geoff Huston
Chief Scientist, APNIC
April 2019

Security on the Internet

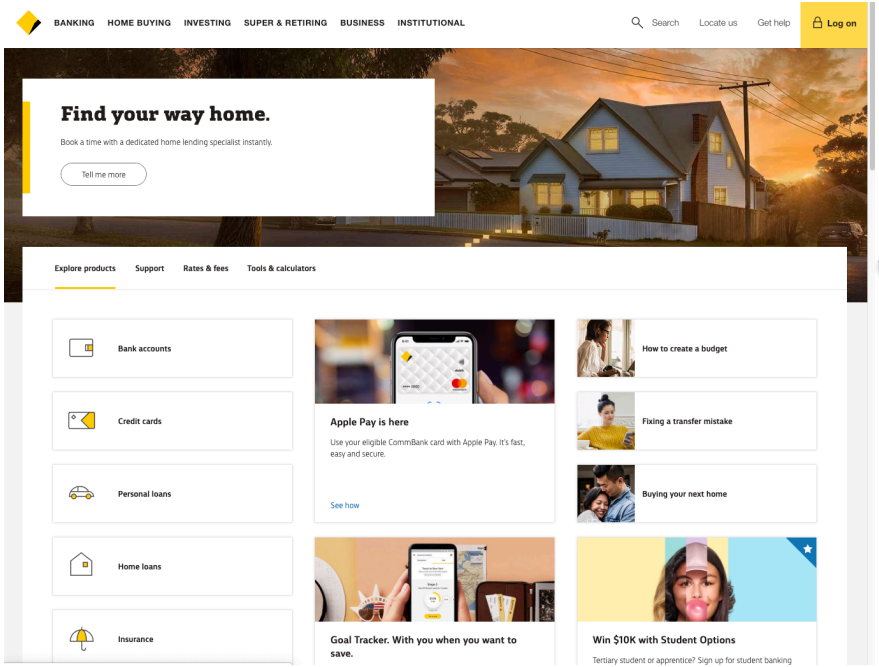
How do you know that you are going to where you thought you were going to?



My bank

Security on the Internet

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

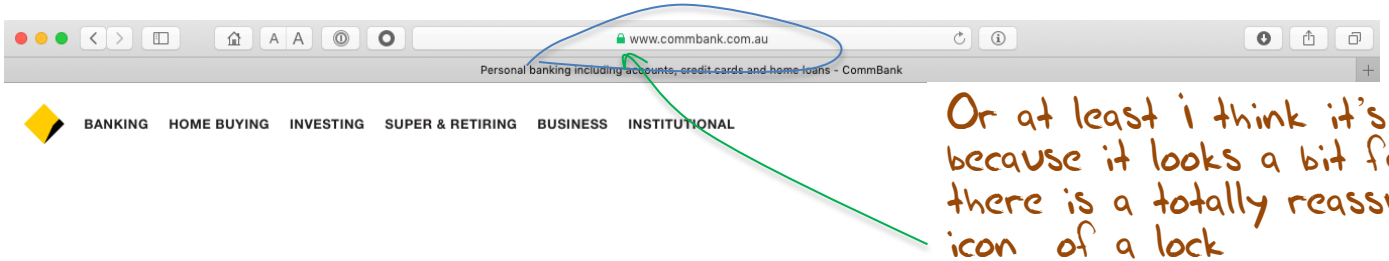


My bank

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.
```

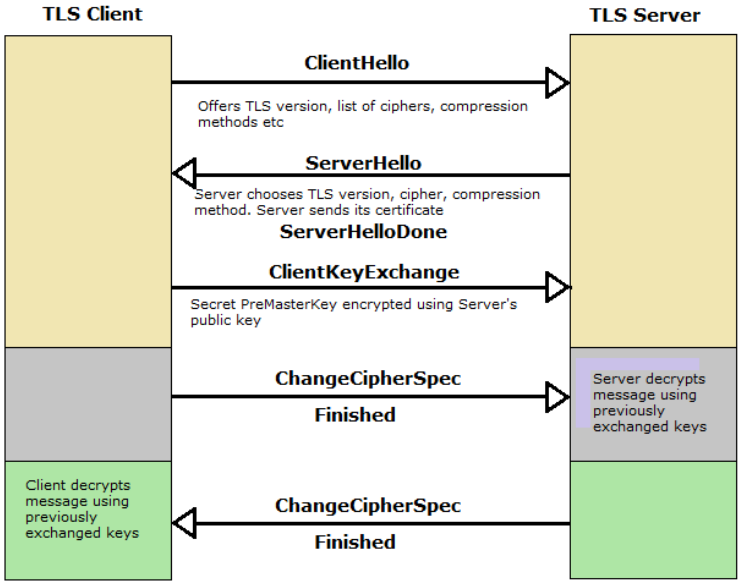
That’s not an IP addresses that was allocated to the Commonwealth Bank!

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?

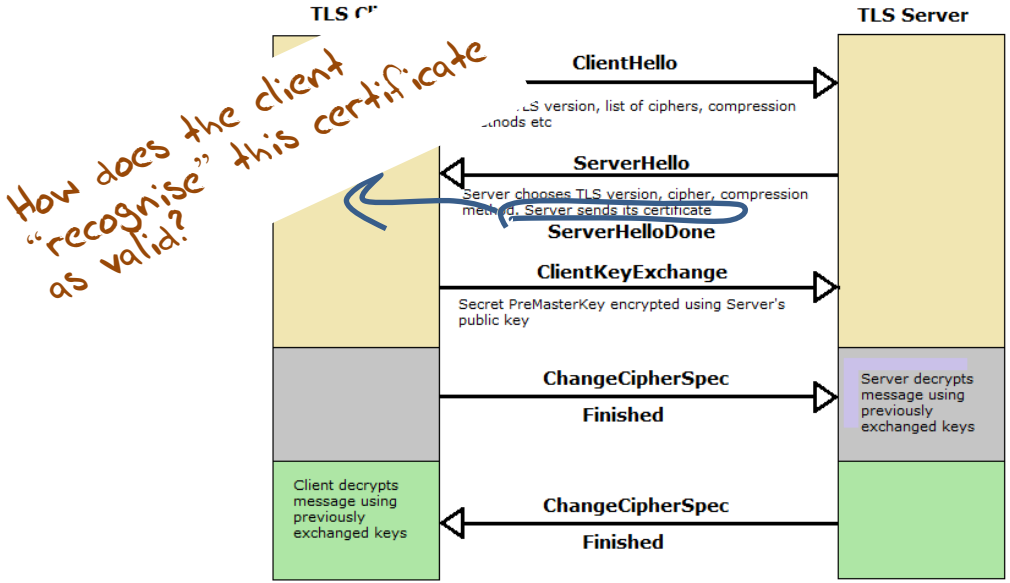
How can my browser tell the difference between an intended truth and a lie?

TCP Port 443 Transport Layer Security (TLS) Connections



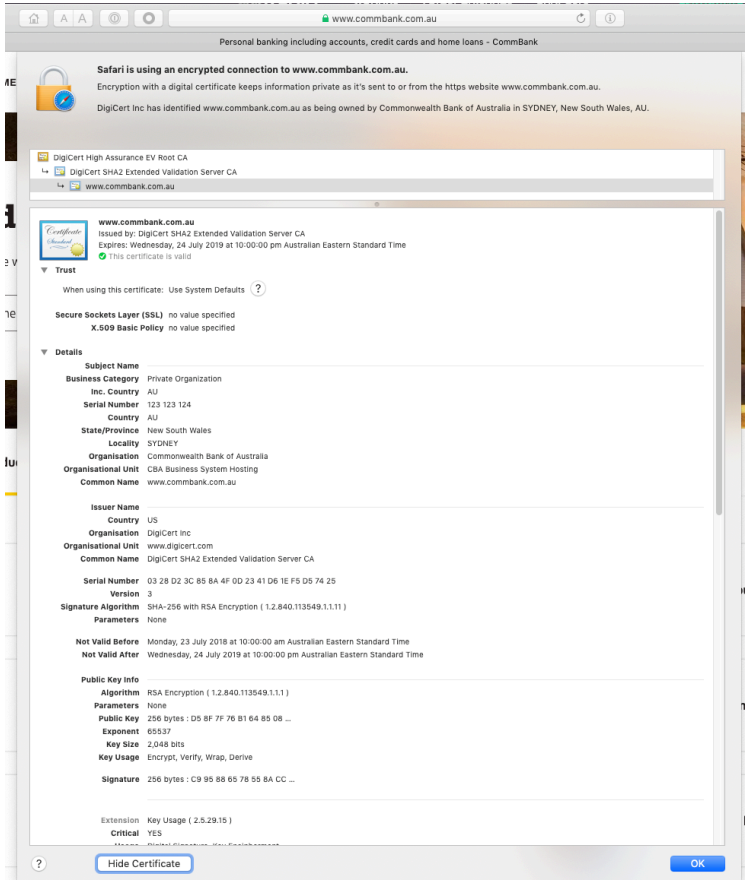
https://rhsecurity.wordpress.com/tag/tls/

TCP Port 443 Transport Layer Security (TLS) Connections

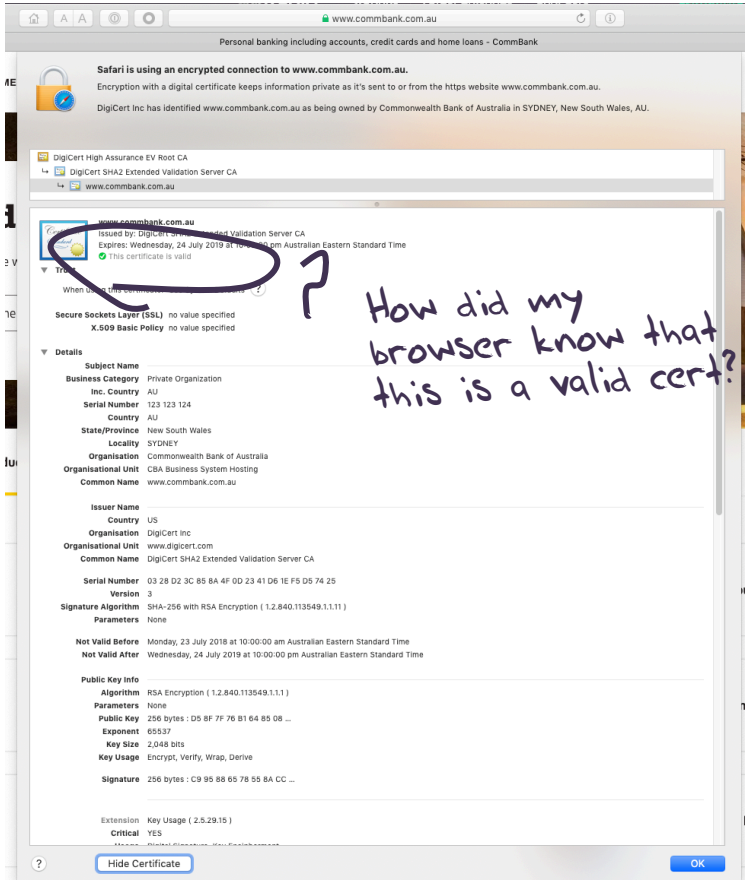


https://rhsecurity.wordpress.com/tag/tls/

The Server's Certificate



The Server's Certificate



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

ABOUTSSL

ABOUT SSL | TYPES OF SSL | SSL WIZARD | HOW TO INSTALL SSL | COMPARE SSL | SSL REVIEWS | SSL TOOLS

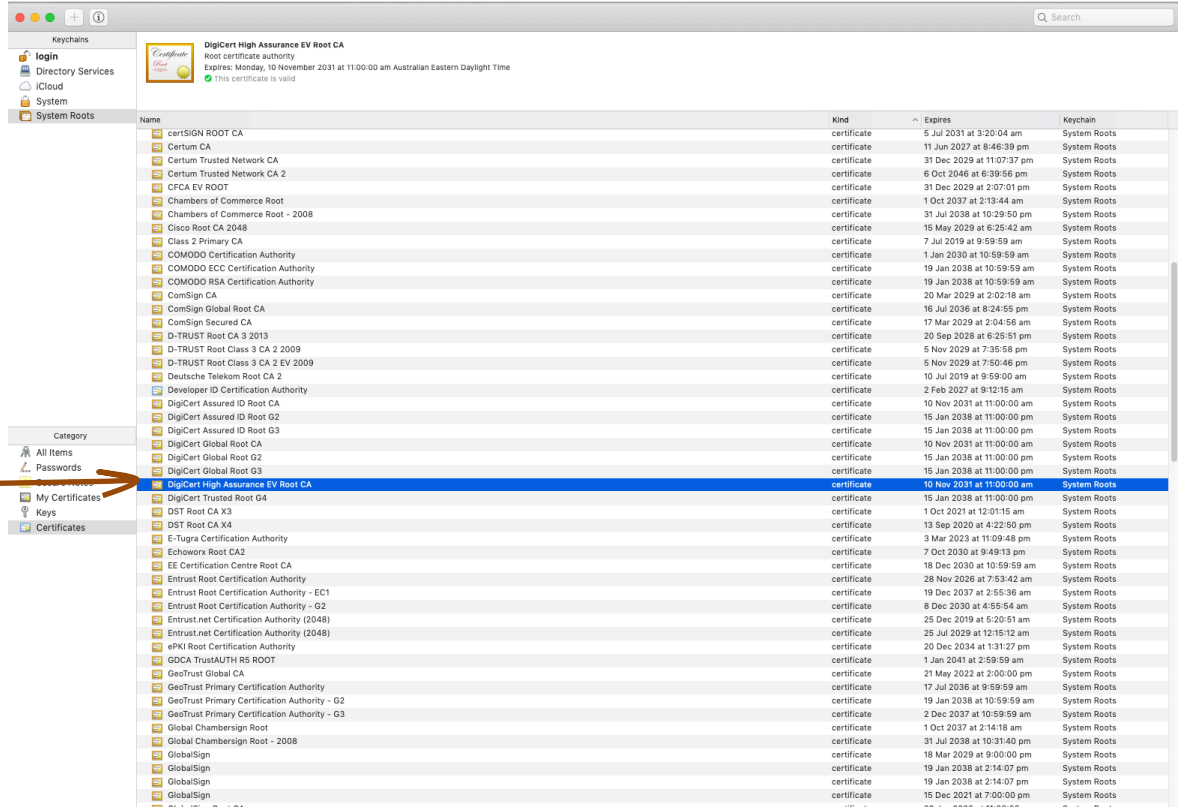
DigiCert Certificate Authority

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**.

digicert

is this the sign of a conscientious CA?

Local Trust

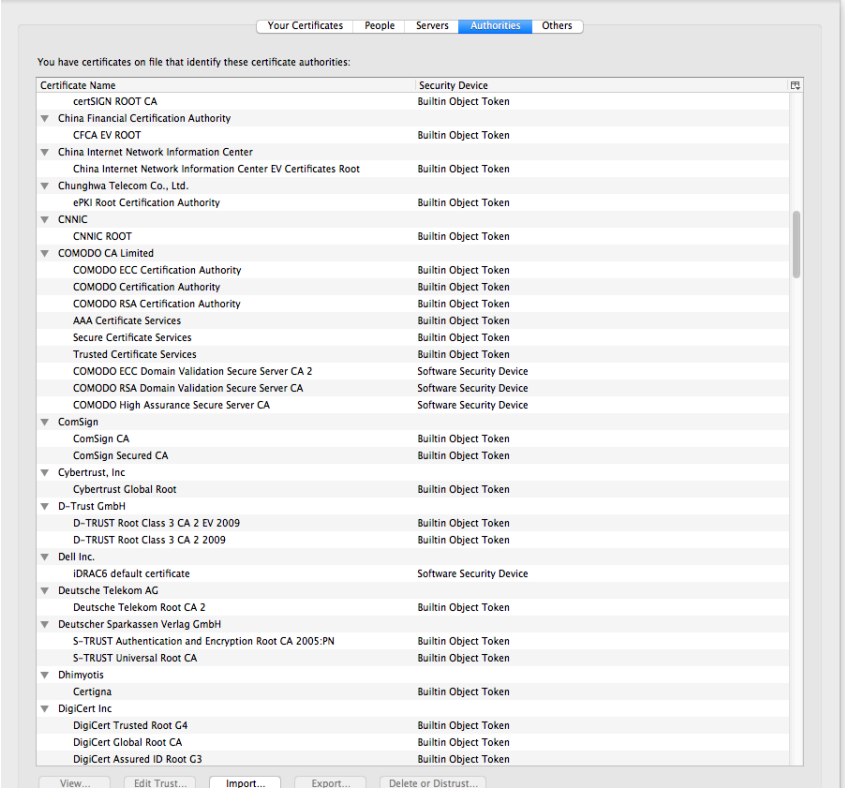


The cert i'm being asked to trust was issued by a certification authority that my browser already trusts – so i trust that cert!

Local Trust or Local Credulity*?

That's a big list of people to Trust

Are they all trustable?



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

Local Credulity

That's a big list of people to Trust

Are they all trustable?

Evidently Not!

Your Certificates | People | Servers | Authorities | Others

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

View... | Edit Trust... | Import... | Export... | Delete or Distrust...

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

Google Online Security Blog: Maintaining digital certificate security

Maintaining digital certificate security

Posted: Monday, March 23, 2015

G+1 106

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!

Your Certificates | People | Servers | Authorities | Others

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

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

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

sters helped ignited
untry's 45-member

talk
ow

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

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

BY SOMINI SENGUPTA

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

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

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

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

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

"I'm totally independent," he said in an e-mail exchange with The New York Times. "I just share my findings with some people in Iran. They are free to do anything they want with my findings and things I share with them, but I'm not resp ible."

In the ternet attacks, this is most ment of reckon

HACKER, PAG

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?

?

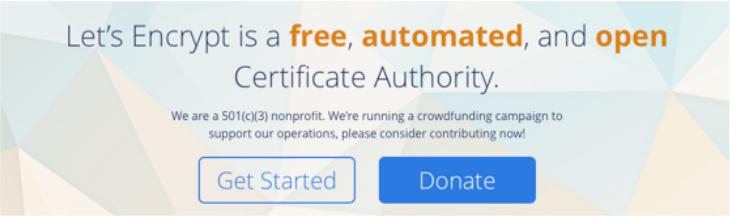
In a Commercial Environment

Where CA's compete with each other for market share

And quality offers no protection

Than what 'wins' in the market?

Cheap Won!



www.letsencrypt.org

Cheap Won!

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

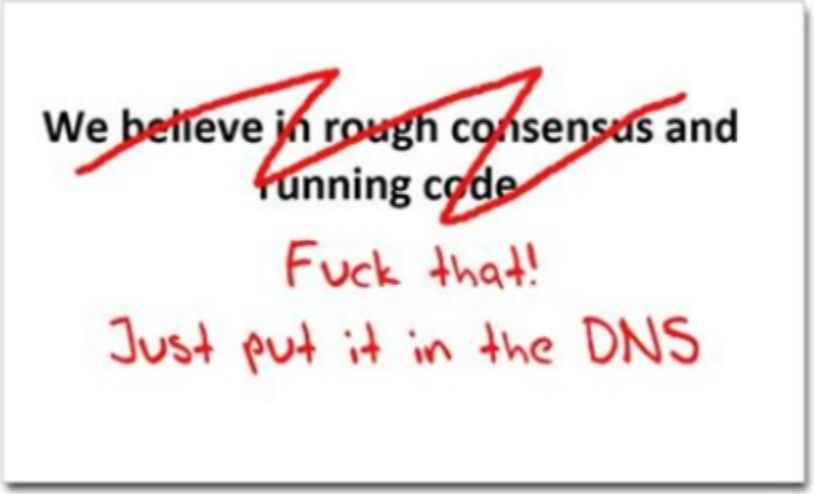
Too little, too slowly

Like the IPv6 transition, we have devised numerous approaches to this problem

But none of them are terribly effective!

Where now?

Use the DNS?



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 issu

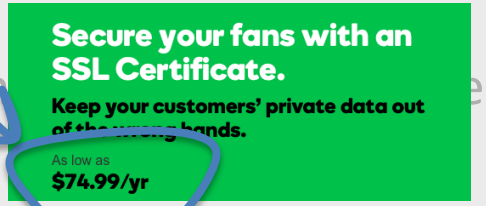
or

- Why not [illegible] for the hash of the domain name cert?

or

- Why not query the DNS for the hash [illegible] e subject public key info?

Who needs CA's anyway?



DANE

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

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

PROPOSED STANDARD

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

V. Dukhovni
Two Sigma
W. Hardaker
Parsons
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

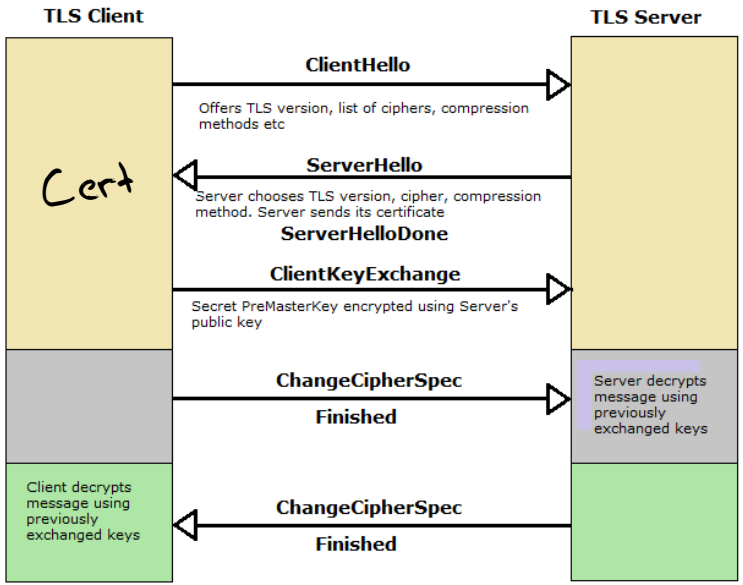
TLS with DANE

- 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

DNS Name

TLSA query



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

. (root)

.com

.co

.example.

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

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?
his record valid?

is the KSK for example.com valid?

example.com Key-Signing Key – signs over

example.com Zone-Signing Key – signs over

www.example.com

is the ZSK for example.com valid?

is the signature for this record valid?

www.example.com IN A 192.0.1

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 <u>all</u> 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!