

Why Dane?

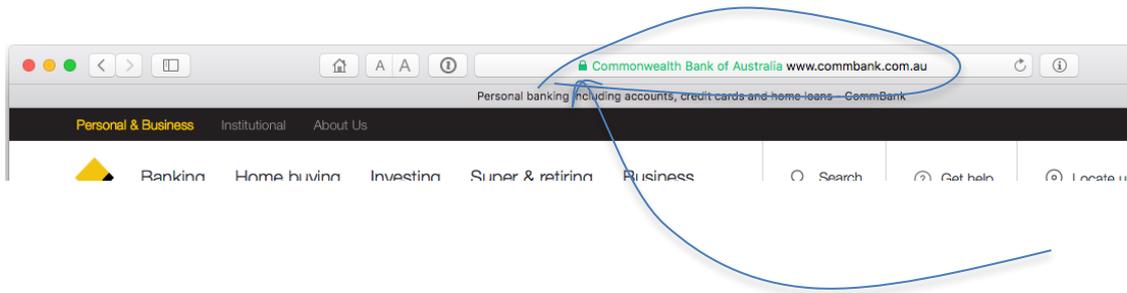
Geoff Huston
Chief Scientist, APNIC

Security on the Internet

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

Security on the Internet

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



My Bank's web site

Or at least i think its my bank because it looks a bit familiar and there is a 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:

104.97.235.12

TCP Session:

TCP Connect 104.97.235.12, port 443



Hang on...

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

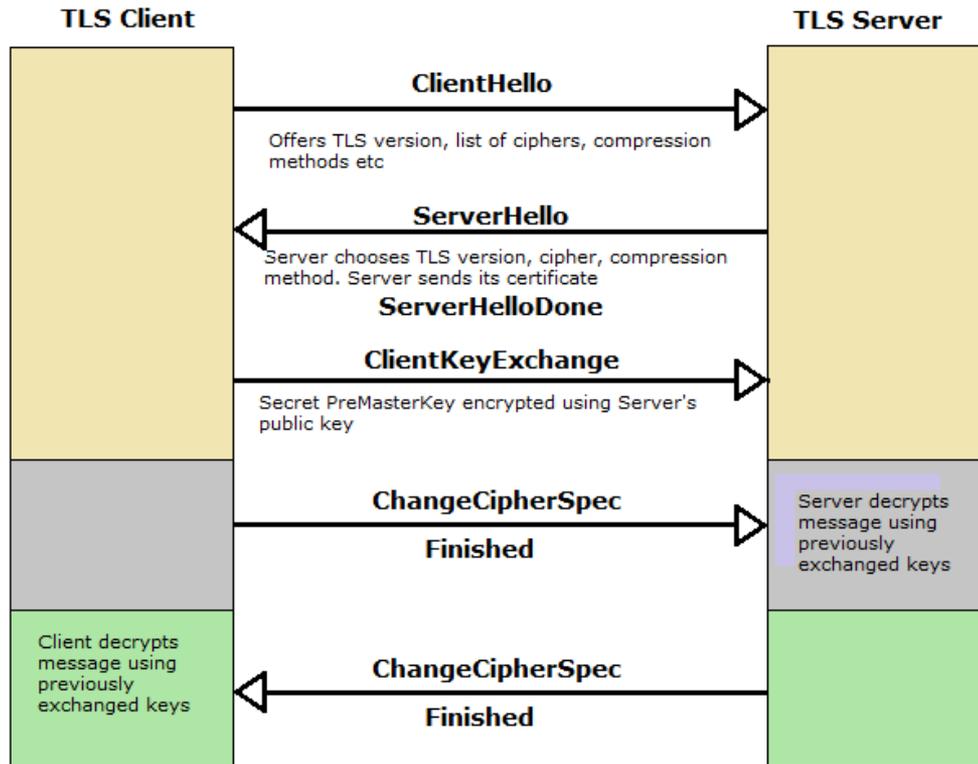
That's not an IP addresses that was allocated to the Commonwealth Bank!

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

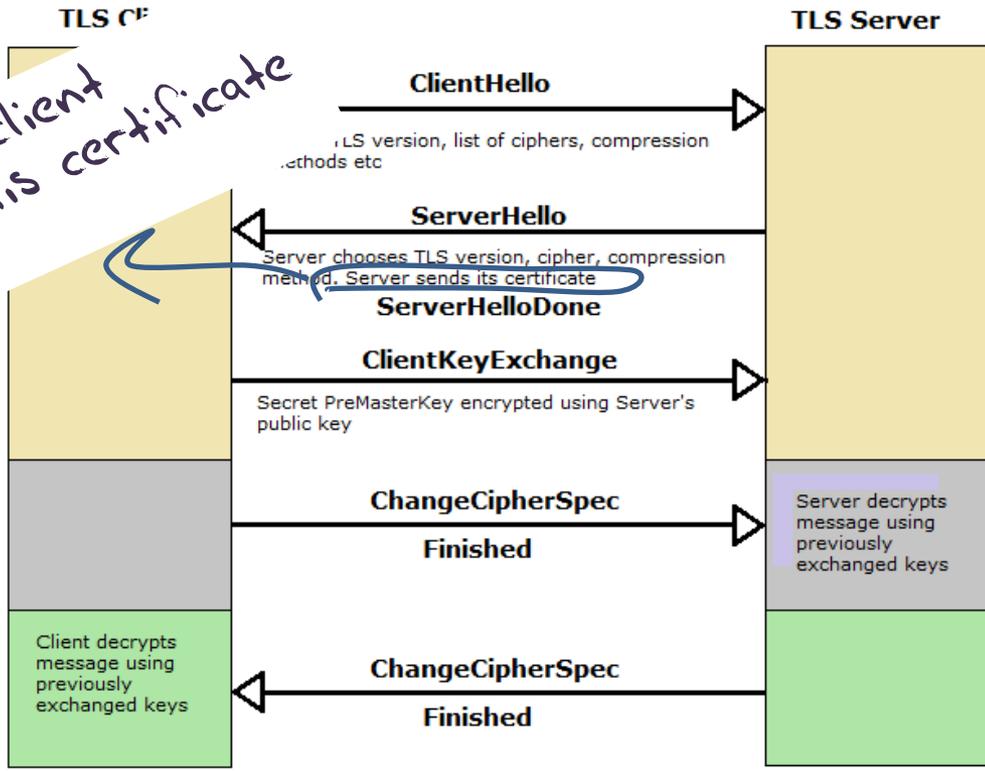
So why should my browser trust that 104.97.235.12 is really the “proper” web site for the Commonwealth Bank of Australia and not some dastardly evil scam?

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

TLS Connections



TLS Connections



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



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 9B AE
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters	none
Not Valid Before	Monday, 4 May 2015 at 10:00:00 AM Australian Eastern Standard Time
Not Valid After	Saturday, 27 February 2016 at 10:59:59 AM Australian Eastern Daylight Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	none
Public Key	256 bytes : CA B4 74 93 E8 00 22 10 ...
Exponent	65537
Key Size	2048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : 95 32 C3 F0 62 F1 F8 F1 ...



Hide Certificate

OK



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?



Hide Certificate

OK

Domain Name Certification

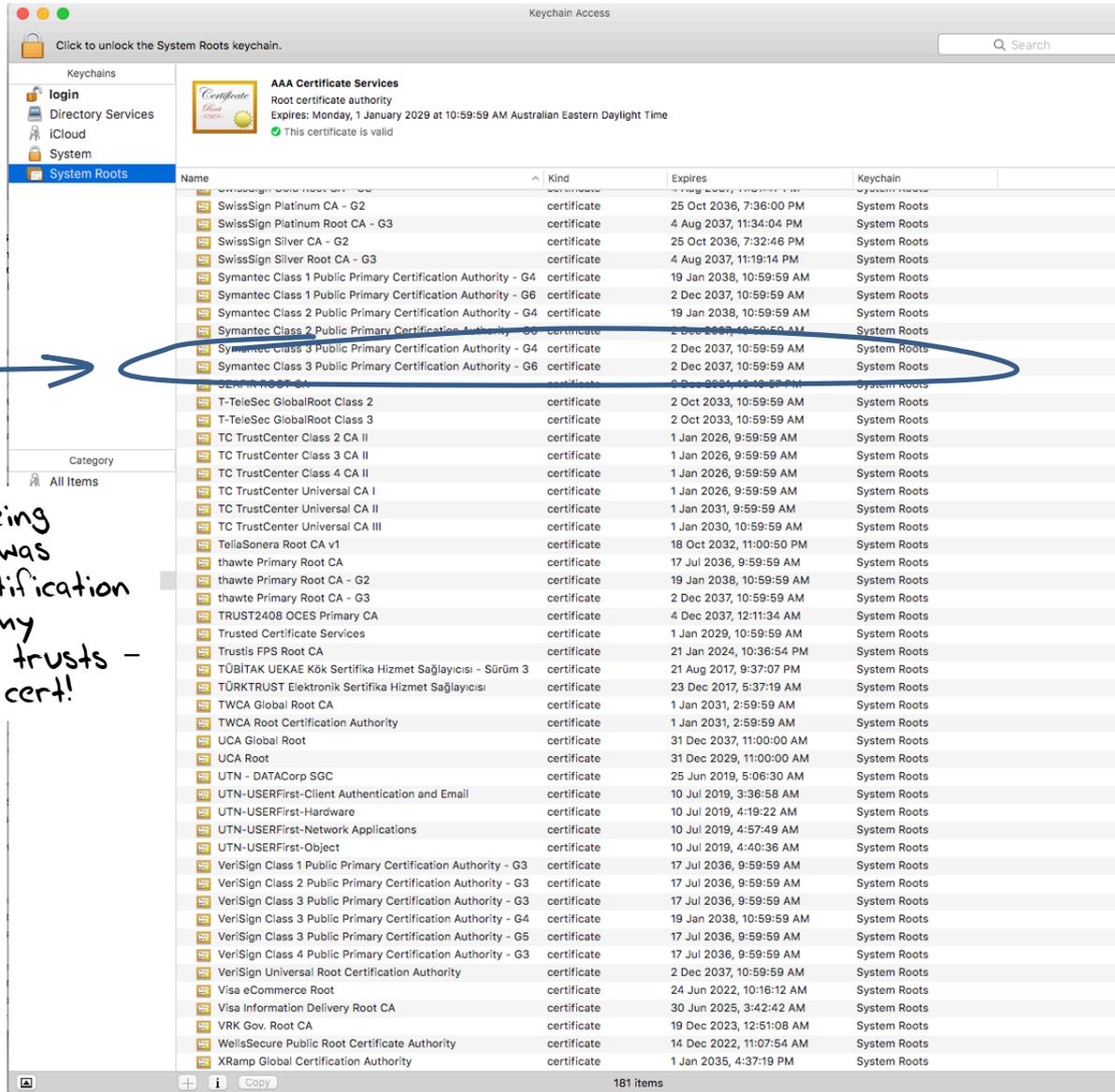
- The Commonwealth Bank of Australia has generated a key pair
- And they passed a Certificate Signing Request to a company called “Symantec” (together with money)
- Symantec is willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value (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 Symantec, 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 “Symantec” (together with money)
- Symantec is willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value (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 Symantec, and their certificate issuance processes, and that the certificates that they issue are always genuine

Why should i trust them?

Local Trust

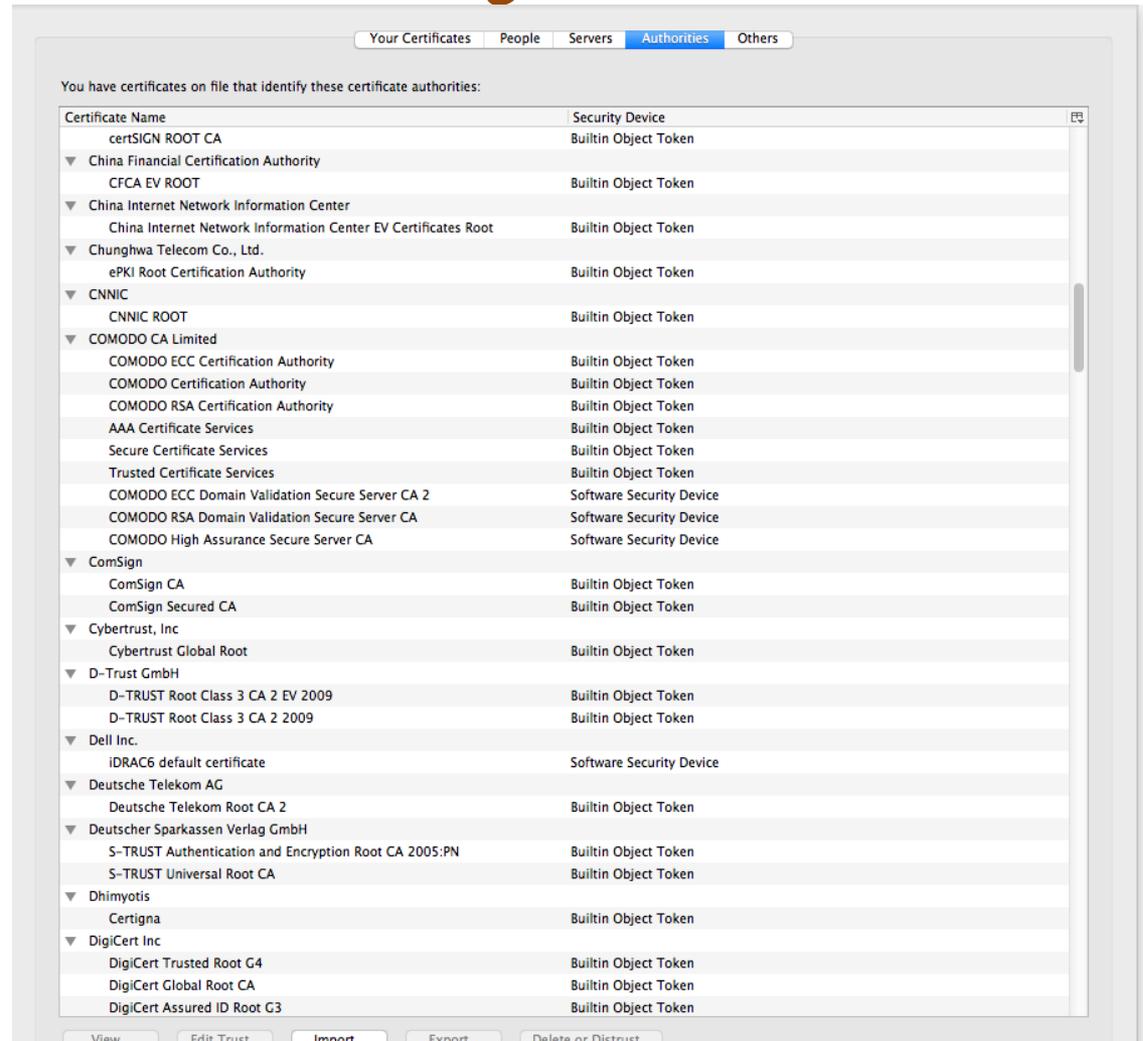


The cert I'm being asked to trust was issued by a certification authority that my browser already trusts - so I trust that cert!

Local Trust or Local Credulity*?

That's a big list of people to Trust

Are they all trustable?



* cre·du·li·ty

/krə'd(y)ooledē/

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!

The image shows a screenshot of a certificate authority list on the left and a blog post on the right. The list includes various CAs such as certSIGN ROOT CA, China Financial Certification Authority, and CNNIC ROOT. A blue circle highlights 'CNNIC ROOT' in the list. The blog post, titled 'Maintaining digital certificate security', discusses an unauthorized digital certificate issued by MCS Holdings, which was trusted by CNNIC. A blue circle highlights a paragraph in the blog post stating that CNNIC is included in all major root stores and that misissued certificates would be trusted by almost all browsers and operating systems.

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
▼ Chunghua Telecom Co., Ltd.	
ePKI Root Certification Authority	Builtin Object Token
▼ CNNIC	
CNNIC ROOT	Builtin Object Token
▼ COMODO CA Limited	
COMODO ECC Certificate Authority	
COMODO Certification Authority	
COMODO RSA Certificate Authority	
AAA Certificate Service	
Secure Certificate Service	
Trusted Certificate Service	
COMODO ECC Domain Authority	
COMODO RSA Domain Authority	
COMODO High Assurance Root	
▼ 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 certificate	
▼ Deutsche Telekom AG	
Deutsche Telekom Root	
▼ Deutscher Sparkassen Verein	
S-TRUST Authenticating Authority	
S-TRUST Universal Root	
▼ Dhimyotis	
Certigna	
▼ DigiCert Inc	
DigiCert Trusted Root	
DigiCert Global Root CA	
DigiCert Assured ID Root G3	Builtin Object Token

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
▼ Chungwa 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...

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

IRE

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

With unpleasant consequences when it all goes wrong

... in the leadership.
... sters helped ignited
... untry's 45-member

... 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

talk
ow

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

BY SOMINI SENGUPTA

Cuba aimed at U.S.
her husband not to
anything happens,
stay right here with
told him in October
to be with you, and I
ou, and the children
without you."

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 sys-
tems of a security firm on the outskirts
of Amsterdam. He created fake creden-
tials 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
include tapping into the online
e-mails of many as 300,000
people last summer.

online security mechanism that is trus-
ted by Internet users all over the world.
Comodohacker, as he calls himself, in-
sists that he acted on his own and is un-
perturbed 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 responsible."

In the
is most
recker,
HACKER, THE

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 digital certificates
WOW! That's awesomely bad!
- 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 a certificate
- That 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!

ANY

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/DefcomSSLiverse.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

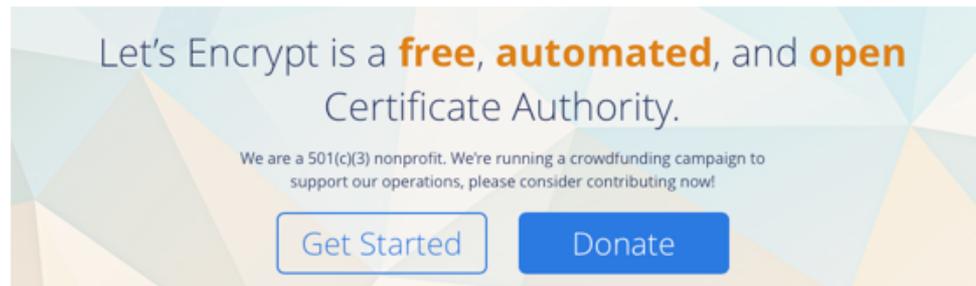
Trusted



Cheap!

Where now?

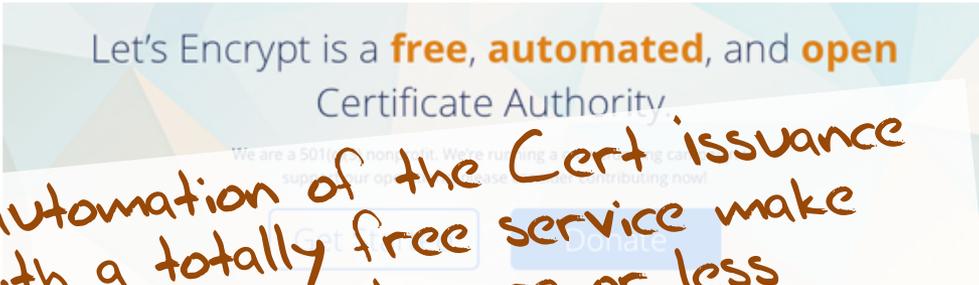
Option A: Take all the money out of the system!



www.letsencrypt.org

Where now?

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



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

We are a 501(c)(3) nonprofit. We're running a public good. We're open to anyone who wants to contribute now!

Get Free Donate

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

www.letsencrypt.org

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

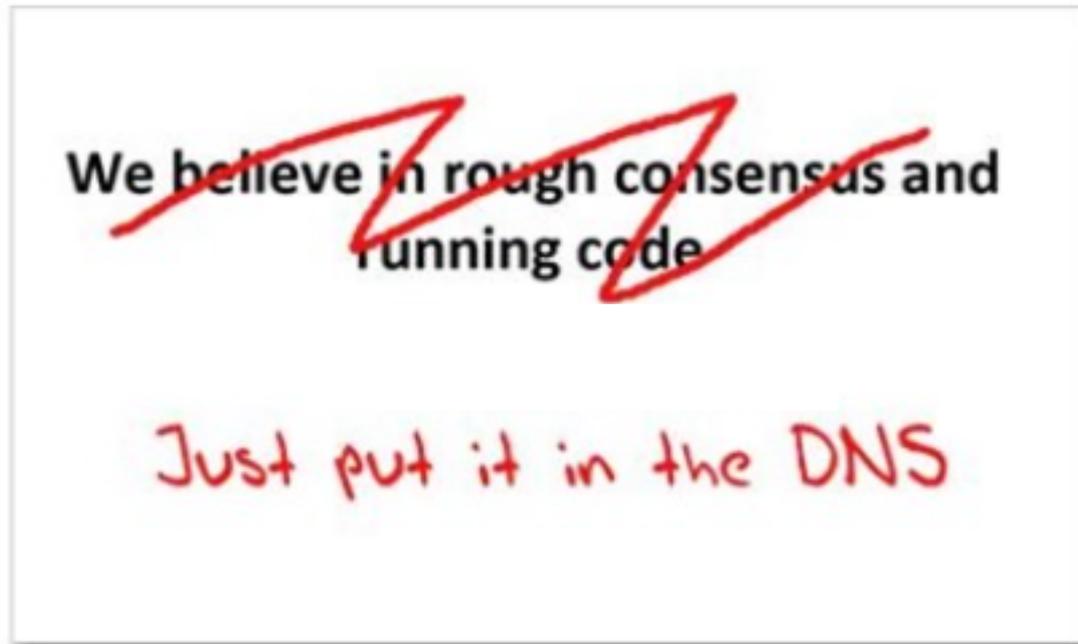
Where now?

Option B: White Listing and Pinning with HSTS

https://code.google.com/p/chromium/src/trunk/net/http/transport_security_state_static.json *its not a totally insane idea -- until you realise that it appears to be completely unscalable!*

Where now?

Option C: Use the DNS!



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?

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 HSTS record (pinning record)?

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 HSTS record?
- Why not query the DNS for the issuer CA?

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

Seriously

Where better to find out the public key associated with a DNS name 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 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?

Who needs CA's anyway?

– 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 name subject public key info?

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

[\[Docs\]](#) [\[txt|pdf\]](#) [\[draft-ietf-dane-p...\]](#) [\[Diff1\]](#) [\[Diff2\]](#) [\[Errata\]](#)

Updated by: [7218](#), [7671](#)

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

PROPOSED STANDARD
[Errata Exist](#)
P. Hoffman
VPN Consortium
J. Schlyter
Kirei AB
August 2012

The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA

Abstract

Encrypted communication on the Internet often uses Transport Layer Security (TLS), which depends on third parties to certify the keys used. This document improves on that situation by enabling the administrators of domain names to specify the keys used in that domain's TLS servers. This requires matching improvements in TLS client software, but no change in TLS server software.

Status of This Memo

This is an Internet Standards Track document.

DANE

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

[\[Docs\]](#) [\[txt|pdf\]](#) [\[draft-ietf-dane-ops\]](#) [\[Diff1\]](#) [\[Diff2\]](#)

PROPOSED STANDARD

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

V. Dukhovni
Two Sigma
W. Hardaker
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  
      7983ald16e8a410e4561cb106618e971 )
```

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:

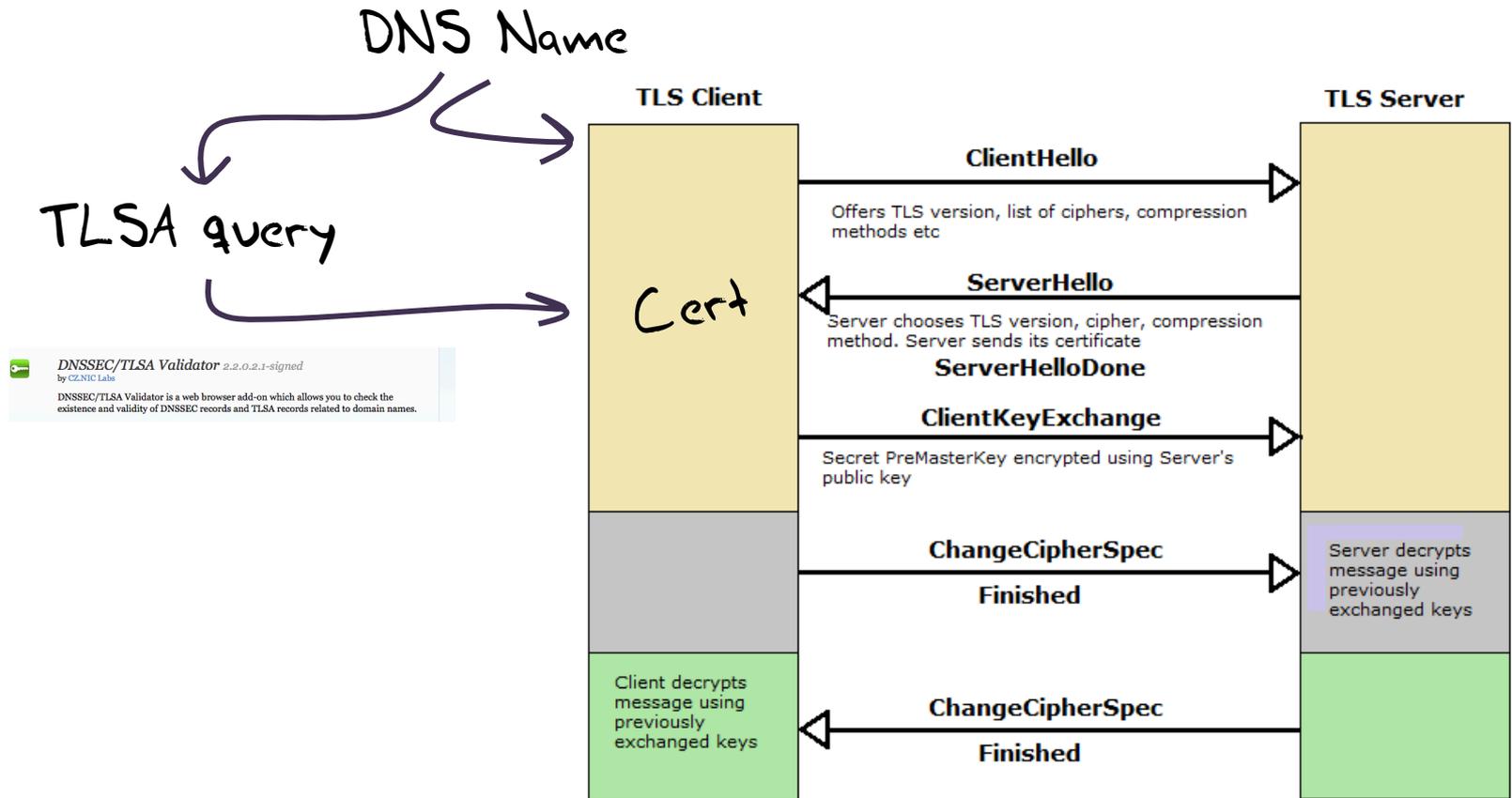
```
_443._tcp.www.example.com. IN TLSA  
  2 0 0 30820307308201efa003020102020... )
```

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



Just one problem...

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

Just one response...

- We need to allow users to validate DNS responses for themselves
- And for this we need a Secure DNS framework
- Which we have – and its called DNSSEC!

DNSSEC Interlocking Signatures

. (root)

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

.com

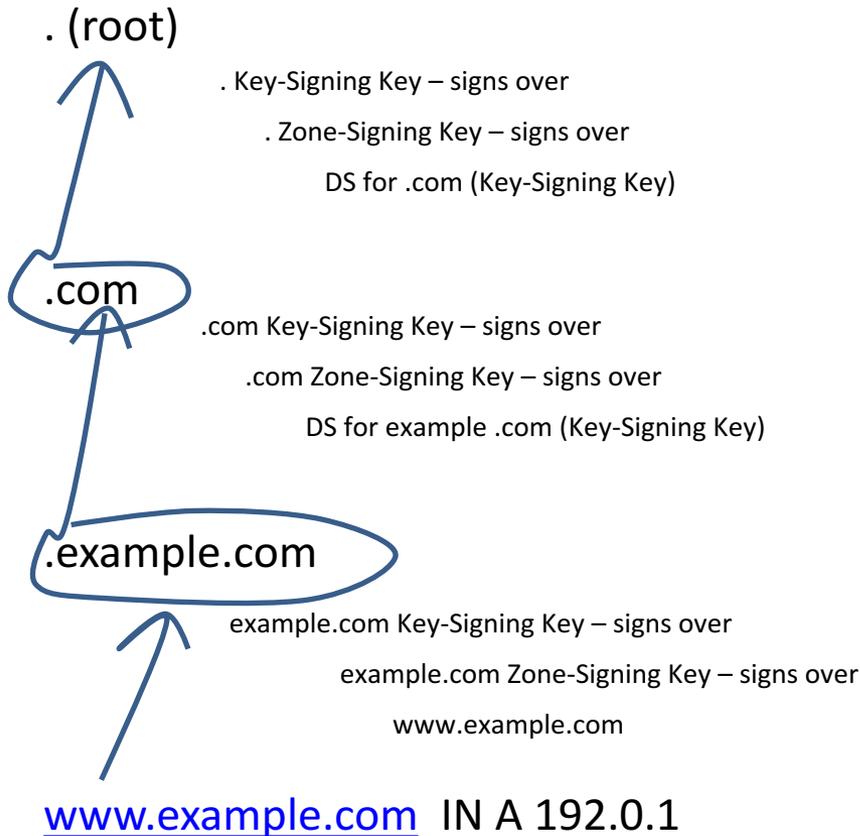
- .com Key-Signing Key – signs over
 - .com Zone-Signing Key – signs over
 - DS for example .com (Key-Signing Key)

.example.com

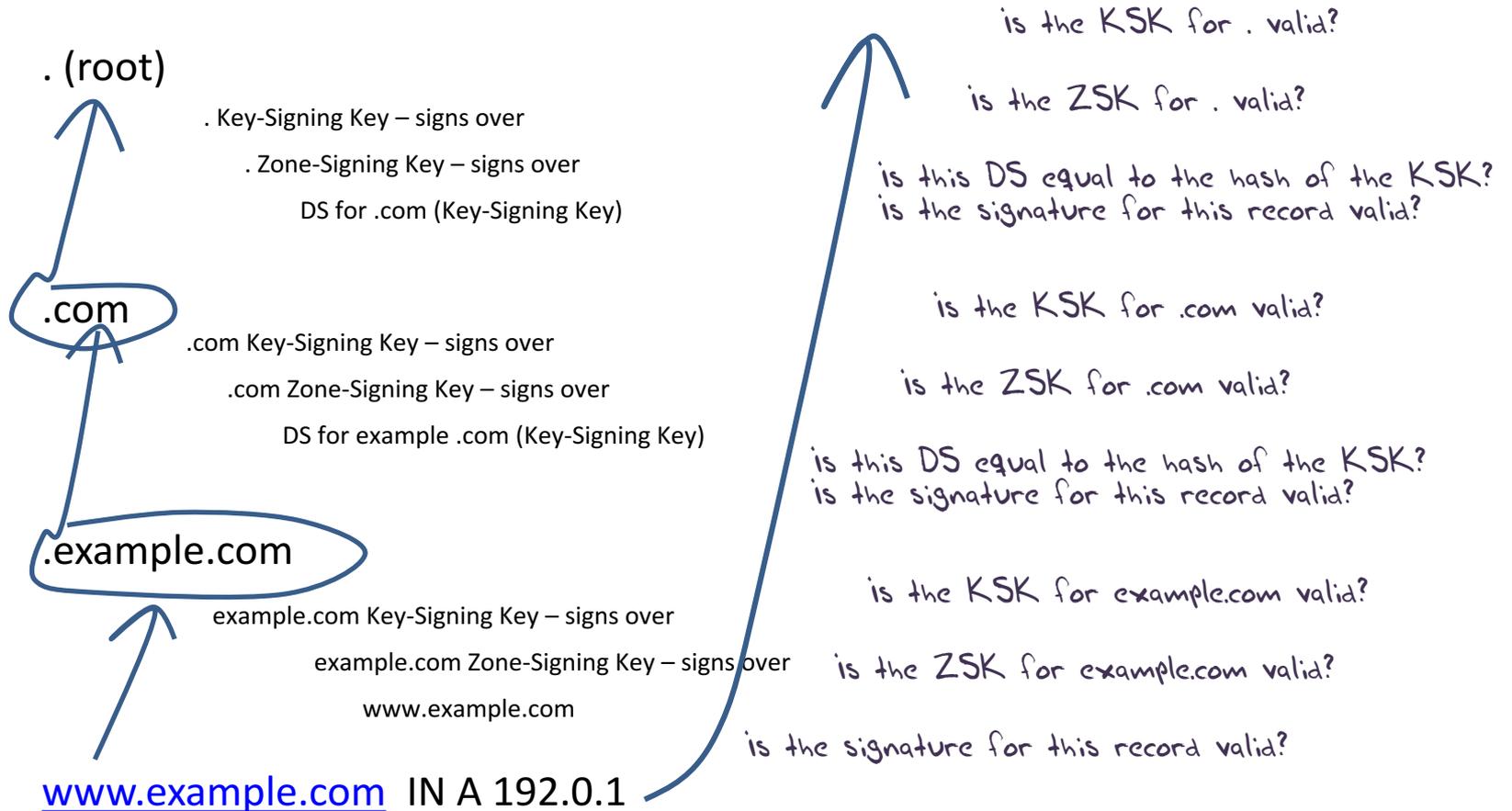
- example.com Key-Signing Key – signs over
 - example.com Zone-Signing Key – signs over
 - www.example.com

www.example.com

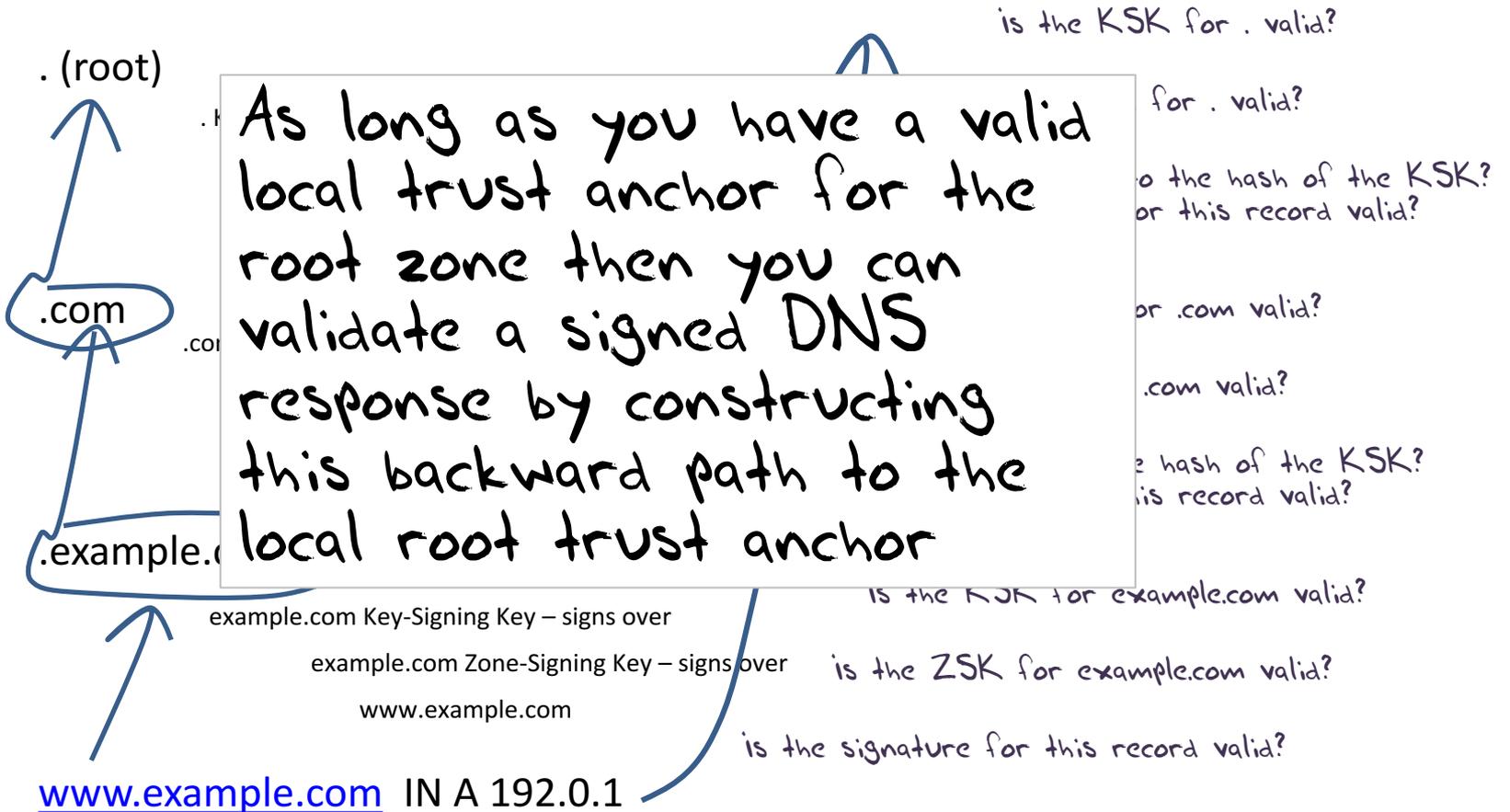
DNSSEC Interlocking Signatures



DNSSEC Interlocking Signatures



DNSSEC Interlocking Signatures



DANE + DNSSEC

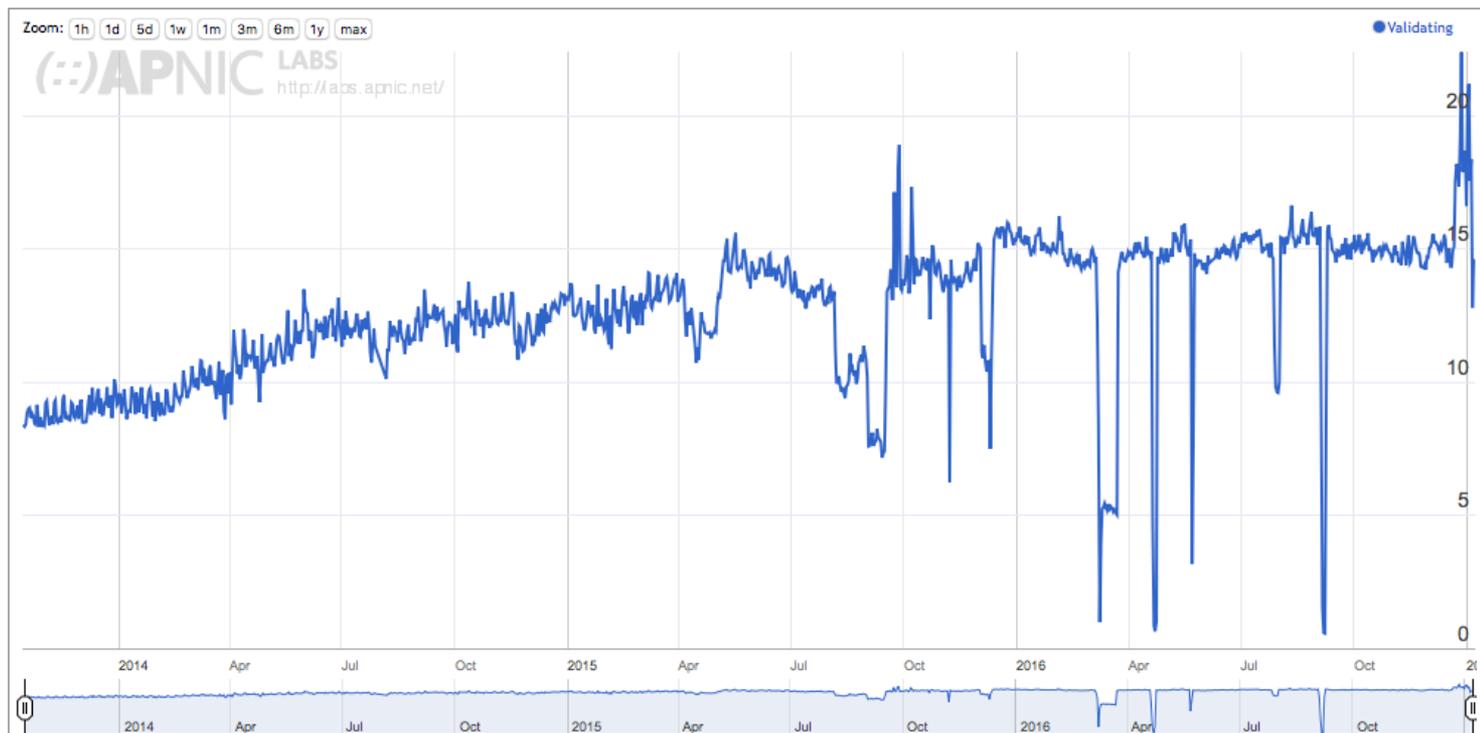
- Query the DNS for the TLSA record of the domain name and ask for the DNSSEC signature to be included in the response
- Validate the signature to ensure that you have an unbroken signature chain to the root trust point
- At this point you can accept the TLSA record as the authentic record, and set up a TLS session based on this data

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

Or...

Look! No DNS!

- Server packages server cert, TLSA record and the DNSSEC credential chain in a single bundle
- Client receives bundle in Server Hello
 - *Client performs validation of TLSA Resource Record using the supplied DNSEC signatures plus the local DNS Root Trust Anchor without performing any DNS queries*
 - *Client validates the presented certificate against the TLSA RR*
- Client performs Client Key exchange

Where now?

Browser vendors appear to be dragging the chain on DANE support

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

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

Maybe this will change

Bug 672600 - Use DNSSEC/DANE chain stapled into TLS handshake in certificate chain validation [Last Comment](#)

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

Mozilla Bug Report 672600

Where now?

We could do a **far** better job at Internet Security:

- Publishing DNSSEC-signed zones

- Publishing DANE TLSA records

- Using DNSSEC-validating resolution

- Using TLSA records to guide Key Exchange for TLS

Let's Do it!



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 compromiseable 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 DNSSEC 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?