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?
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

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 Client**

- **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 PremasterSecret encrypted using Server's public key

**TLS Server**

- **ChangeCipherSpec**

- **Finished**
  - Server decrypts message using previously exchanged keys

- **ChangeCipherSpec**

- **Finished**
  - Client decrypts message using previously exchanged keys
TLS Connections

How does the client "recognise" this certificate as valid?
How did my browser know that this is a valid cert?
Domain Name Certification

• The Commonwealth Bank of Australia has generated a key pair

• And they passed a certificate signing request to a company called “Symantec”

• Who is willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value

• So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to www.commbank.com.au, as long as I am prepared to trust Symantec and the certificates that they issue
Domain Name Certification

• The Commonwealth Bank of Australia has generated a key pair

• And they passed a certificate signing request to a company called “Symantec”

• Who is willing to vouch (in a certificate) that the entity who goes by the domain name of www.commbank.com.au also has a certain public key value

• So if I can associate this public key with a connection then I have a high degree of confidence that I’ve connected to www.commbank.com.au, as long as I am prepared to trust Symantec and the certificates that they issue

Why should I trust them?
Local Trust

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

That’s a big list of people to Trust

Are they all trustable?

<table>
<thead>
<tr>
<th>Certificate Name</th>
<th>Security Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>certSIGN ROOT CA</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>China Financial Certification Authority</td>
<td></td>
</tr>
<tr>
<td>CFCA EV ROOT</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>China Internet Network Information Center</td>
<td></td>
</tr>
<tr>
<td>China Internet Network Information Center EV Certificates Root</td>
<td></td>
</tr>
<tr>
<td>Chunghwa Telecom Co., Ltd.</td>
<td></td>
</tr>
<tr>
<td>ePKI Root Certification Authority</td>
<td></td>
</tr>
<tr>
<td>CNNIC</td>
<td></td>
</tr>
<tr>
<td>CNNIC ROOT</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>COMODO CA Limited</td>
<td></td>
</tr>
<tr>
<td>COMODO ECC Certification Authority</td>
<td></td>
</tr>
<tr>
<td>COMODO Certification Authority</td>
<td></td>
</tr>
<tr>
<td>COMODO RSA Certification Authority</td>
<td></td>
</tr>
<tr>
<td>AAA Certificate Services</td>
<td></td>
</tr>
<tr>
<td>Secure Certificate Services</td>
<td></td>
</tr>
<tr>
<td>Trusted Certificate Services</td>
<td></td>
</tr>
<tr>
<td>COMODO ECC Domain Validation Secure Server CA 2</td>
<td>Software Security Device</td>
</tr>
<tr>
<td>COMODO RSA Domain Validation Secure Server CA</td>
<td>Software Security Device</td>
</tr>
<tr>
<td>COMODO High Assurance Secure Server CA</td>
<td>Software Security Device</td>
</tr>
<tr>
<td>CodeSign</td>
<td></td>
</tr>
<tr>
<td>CodeSign CA</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>CodeSign Secured CA</td>
<td></td>
</tr>
<tr>
<td>Cybertrust, Inc</td>
<td></td>
</tr>
<tr>
<td>Cybertrust Global Root</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>D-Trust GmbH</td>
<td></td>
</tr>
<tr>
<td>D-TRUST Root Class 3 CA 2 EV 2009</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>D-TRUST Root Class 3 CA 2 2009</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>Dell Inc</td>
<td></td>
</tr>
<tr>
<td>IDIACs default certificate</td>
<td>Software Security Device</td>
</tr>
<tr>
<td>Deutsche Telekom AG</td>
<td></td>
</tr>
<tr>
<td>Deutsche Telekom Root CA 2</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>Deutscher Sparkassen Verlag GmbH</td>
<td></td>
</tr>
<tr>
<td>S-TRAU Authentication and Encryption Root CA 2005 PN</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>S-TRAU Universal Root CA</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>Dhinymds</td>
<td></td>
</tr>
<tr>
<td>Certigna</td>
<td></td>
</tr>
<tr>
<td>DigiCert Inc</td>
<td></td>
</tr>
<tr>
<td>DigiCert Trusted Root G4</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>DigiCert Global Root CA</td>
<td>Built-in Object Token</td>
</tr>
<tr>
<td>DigiCert Assured ID Root G3</td>
<td>Built-in Object Token</td>
</tr>
</tbody>
</table>
Local Trust

That’s a big list of people to Trust

Are they all trustable?

Evidently Not!
Local Trust

That's a big list of people to Trust

Are they all trustable?  

Evidently Not!
With unpleasant consequences when it all goes wrong
With unpleasant consequences when it all goes wrong

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 have been used to tap into the online communications of as many as 300,000 people this summer.

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

International Herald Tribune

Sep 13, 2011 Front Page
With unpleasant consequences when it all goes wrong.
What's going wrong here?

• The TLS handshake cannot specify WHICH CA should be used to validate the digital certificate

• Your browser will allow ANY CA to be used to validate a certificate
What's going wrong here?

• The TLS handshake cannot specify WHICH CA should be used to validate the digital certificate.

• Your browser allows 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.
- Your browser allows ANY CA to be used to validate a certificate.

WOW! That's awesomely bad!

Here's a lock - it might be the lock on your front door for all I know.

The lock might LOOK secure, but don't worry - literally ANY key can open it!
What's going wrong here?

• There is no incentive for quality in the CA marketplace
• Why pay more for any certificate when the entire CA structure is only as strong as the weakest CA
• And you browser trusts a LOT of CAs!
  – About 60 – 100 CA’s
  – About 1,500 Subordinate RA’s
  – Operated by 650 different organisations

See the EFF SSL observatory
http://www.eff.org/files/DefconSSLiverse.pdf
In a commercial environment

Where CA’s compete with each other for market share
And quality offers no protection
Than what ‘wins’ in the market?

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!
Where now?

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

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

We're probably going to find out real soon!
Where now?

Option B: White Listing and Pinning with HSTS

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

It's not a totally insane idea -- until you realise that it appears to be completely unscaleable!
Where now?

Option C: Use the DNS!

We believe in rough consensus and running code.

Fuck that!

Just put it 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?
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?
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 domain name public key cert as a simple self-signed 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 domain name public key cert as a simple self-signed cert?
Using the DNS to associated domain name public key certificates with domain name

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.
2.3. TLSA RR Examples

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

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

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

_443._tcp.www.example.com. IN TLSA
  1 1 2 92003ba34942dc74152e2f2c408d29ec
  a5a520e7f2e06bb944f4dca346baf63c
  1b177615d466f6c4b71c216a50292bd5
  8c9ebdd2f74e38fe51f4d48c43326c6c )

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

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

TLS Client

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

TLS Server

Server decrypts message using previously exchanged keys

ChangeCipherSpec

Finished

Client decrypts message using previously exchanged keys

TLSoConnections

https://rhsecurity.wordpress.com/tag/tls/
Just one problem...

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

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

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

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

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

www.example.com
DNSSEC Interlocking Signatures

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

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

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

www.example.com IN A 192.0.1
DNSSEC Interlocking Signatures

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

  .com 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

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

• Query the DNS for the TLSA record of the domain name and ask for the DNSSEC signature to be included in the response

• Validate the signature to ensure that you have an unbroken signature chain to the root trust point

• At this point you can accept the TLSA record as the authentic record, and set up a TLS session based on this data
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 plugins rather than a core functionality

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

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

Questions?