A New Internet Architecture for Secure Key Learning: DANE

Eric Osterweil
Principal Scientist, Verisign Labs
August 6, 2015
Something fundamental has been missing from security protections on the Internet

- Our Internet security has had a loophole for years
  - We have TLS, IPSec, S/MIME, SSH, etc.
  - They give us: privacy, encryption, integrity protection, & more

- Protections are mature, have extensive codebases, and are well understood

- But, almost all of them lack important protections during their startup phases (secure bootstrapping)
Examples of creating secure connections today (w/o DANE)

- Sending/receiving secure Inter-Administrative email (e.g. S/MIME)
  - We use out-of-band key bootstrapping to learn keys (user by user b/c we need to know ID-to-key bindings ahead of time). Then we lookup a mail domain in DNS, connect to and ask a server about an email identity and do verif/decrypting/etc w/ pre-learned keys
  - Because we can’t securely learn the keys without out-of-band trust
Examples of creating secure connections today (w/o DANE)
Examples of creating secure connections today (w/o DANE)

- Connecting to secure websites (e.g. HTTPS over TLS)
  - We use out-of-band key bootstrapping to get a list of globally trusted CA keys. Then we lookup a website’s IP address(es) through DNS, fetch a crypto key over an insecure TCP connection, and validate its key using CA keys
  - We learn CA keys using out-of-band trust
Examples of creating secure connections today (w/o DANE)

I want to securely visit example.com

DigiNotar?

CAs
Examples of creating secure connections today (w/o DANE)

What’s missing is secure *key learning*
DANE uses DNSSEC for secure key learning

- **DANE**: DNS-based Authentication of Named Entities

- DANE is an *architectural* substrate for Internet key learning in
  - TLS, S/MIME, PGP, IPSec, etc.

- Don’t do key learning *after* DNS, do it *with* DNS

- DANE is the killer app for DNSSEC

- DANE aligns costs with incentives so that there’s a reason to *ASK* for DNSSEC!
Outline

• A brief overview of DNSSEC (DANE’s substrate)

• How DANE works

• Security using DANE

• Verification with the WebPKI and with DANE

• Examples of DANE protocols and open tools
Why we need DNSSEC

• DNS cache poisoning has been a known attack against DNS since the 1990s [1]

• DNSSEC was designed to cryptographically ensure data’s origin authenticity and integrity

• Then came the “summer of fear” – ’08
  • The Kaminsky attack
  • Patches (source port randomization) helped in the short-term

How DNSSEC works

- First attempt to secure a core Internet protocol w/ crypto

- DNSSEC zones create pub/priv keys
  - Public key is DNSKEY

- Zones sign all RRsets and resolvers use DNSKEYs to verify them
  - Each RRset has a signature attached to it: RRSIG

- Resolvers are configured with a *single root* key, and *all* trust flows recursively down the hierarchy
Data Signing Example

Using a zone’s key on a standard RRset (the NS)

Signature (RRSIG) will only verify with the DNSKEY if no data was modified
**DNSSEC: Validating**

A *Validating Recursive Resolver* uses the root's public key to verify (validate) delegations.
DANE’s architecture
Application uses of DANE

• Allow applications to securely obtain (authenticate) those keys and use them in application security protocols

• Some possible applications: SSH, SSL/TLS, HTTPS, S/MIME, PGP, SMTP, DKIM, and many others..

• DANE records:
  • TLSA
  • Upcoming: OPENPGPKEY, SMIMEA, IPSECA, …

• DANE-like legacy records:
  • SSHFP, IPSECKEY, DKIM TXT record, …
Security for TLS: Using DANE

I want to securely visit example.com

ROOT SERVER

Key Information is learned securely from DNSSEC

CHAIN OF TRUST

This simple record uniquely describes a key, and DNSSEC ensures it's authentic

example.com NAME SERVER

This is the key you will see when you connect
Security for TLS, Using CAs and DANE

I want to securely visit example.com

When browser connects, DANE has already identified the key

This is the key you will see when you connect

Verisign Public
Security for email, using DANE
Without DANE, we have used the WebPKI

- Applications have needed to trust a large number of global Certification Authorities (CA)
- No namespace constraints! Any CA can issue certificates for any entity on the Internet
  - “An attack on one defeats all” [2]
  - Least common denominator security: our collective security is equal to the weakest one!
- Furthermore, many of them issue subordinate CA certificates to their customers, again with no naming constraints

WebPKI model issues

  - Over 1,800 separate CAs are capable of issuing certificates for anyone! (Root CAs and intermediate CAs issued by them)

- “The Shape & Size of Threats: Defining a Networked System’s Attack Surface”
  - Eric Osterweil (Verisign), Danny McPherson (Verisign), Lixia Zhang (UCLA), NPsec 2014 best paper
WebPKI Verification

- Transport Layer Security (TLS) needs to be bootstrapped by cryptographic keys
- CA verification uses a set of globally trusted authorities who can *each* vouch for *any* certificate’s authenticity
  - Certificates represent previous verification: contain signatures from CAs, and point to revocation points for status checks
DANE verification process

- DNS-Based Authentication of Named Entities (DANE)
  - IETF working group, and standards track RFC for TLS
- DNS zones have TLSA record(s) that uniquely authorize cert used by web servers
Look at what we just cut out…

- Qualitatively, a picture is worth 1,000 words: we can see that the attack surface is reduced
- By cutting out our WebPKI check and revocation checks, we removed a lot of moving parts
Count what we just cut out (Alexa 1,000) [3]…

DANE Protocols and Tools
What it takes for DANE to work
Secure Provisioning: A Proof of Concept Portal

- Free provisioning web UI and REST API
- Limited RR types (DANE focused)
- Users can change their keys without affecting parent zone
Experimental Service to encourage adoption

**Provisioning Portal**

**Provisioning Portal Documentation**

https://www.dane-provisioning.verisignlabs.com
DANE for TLS

- RFC 6698: The **DNS-based Authentication of Named Entities (DANE)** Protocol for Transport Layer Security
- Defines a new DNS record type “**TLSA**”, that can be used for better & more secure ways to authenticate SSL/TLS certificates
  - By specifying constraints on which CA can vouch for a certificate, or which specific PKIX end-entity certificate is valid
  - By specifying that a service certificate or a CA can be directly authenticated in the DNS itself.
TLSA record example

_port, transport proto & server domain name_

_TLSA rrtype_

_usage_

_selector_

_matching type_

_certificate association data_

_443._tcp.www.example.com. IN TLSA ( 0 0 1 d2abde240d7cd3ee6b4b28c54df034b9 7983a1d16e8a410e4561cb106618e971 )_
TLSA configuration parameters

**Usage field:**

- 0 PKIX-TA: CA Constraint
- 1 PKIX-EE: Service Certificate Constraint
- 2 DANE-TA: Trust Anchor Assertion
- 3 DANE-EE: Domain Issued Certificate

**Selector field:**

- 0 Match full certificate
- 1 Match only SubjectPublicKeyInfo

**Matching type field:**

- 0 Exact match on selected content
- 1 SHA-256 hash of selected content
- 2 SHA-512 hash of selected content

Certificate Association Data: raw cert data in hex
TLSA configuration parameters

**Usage field:**

- 0  PKIX-TA: CA Constraint
- 1  PKIX-EE: Service Certificate Constraint
- 2  DANE-TA: Trust Anchor Assertion
- 3  DANE-EE: Domain Issued Certificate

**Selector field:**

- 0  Match full certificate
- 1  Match only SubjectPublicKeyInfo

**Matching type field:**

- 0  Exact match on selected content
- 1  SHA-256 hash of selected content
- 2  SHA-512 hash of selected content

Certificate Association Data: raw cert data in hex
Usage types

0 **PKIX-TA: CA Constraint**
Specify which CA should be trusted to authenticate the certificate for the service. Full PKIX certificate chain validation needs to be performed.

1 **PKIX-EE: Service Certificate Constraint**
Define which specific service certificate ("EE cert") should be trusted for the service. Full PKIX cert validation needs to be performed.

2 **DANE-TA: Trust Anchor Assertion**
Specify a domain operated CA which should be trusted independently to vouch for the service certificate.

3 **DANE-EE: Domain Issued Certificate**
Define a specific service certificate for the service at this domain name.
Example TLSA record (for WWW)

_443._tcp.fedoraproject.org. 263 IN TLSA 0 0 1 (19400BE5B7A31FB733917700789D2F0A2471C0C9D506C0E504C06C16D7CB17C0)

Usage 0 ("CA Constraint") – this record says:
- For service at fedoraproject.org tcp port 443
- only the CA with the specified SHA-256 certificate fingerprint (19400BE5B...) should be trusted
DANE/TLSA tools and software

• TLSA Record Generation
  • Command line tools: “tlsagen” (in libsmaug), “swede”, “hash-slinger”, “ldns-dane”
  • Web based tool: https://www.huque.com/bin/gen_tlsa

• TLSA validators for web
  • Some 3rd party validator plugins are available (Firefox, Chrome, Opera, Safari):
    • https://www.dnssec-validator.cz/
    • http://blog.huque.com/2014/02/dnssec-dane-tlsa-browser-addons.html
  • Bloodhound Mozilla fork:
    • https://www.dnssec-tools.org/wiki/index.php/Bloodhound
SMIMEA

- Using DNSSEC to associate certificates with domain names for S/MIME
- S/MIME is a method of encrypted and signing MIME data used in email messages
- The SMIMEA DNS record proposes to associate S/MIME certificates with DNS domain names
- Verisign DANE/SMIMEA early Mail User Agent Prototype
Object Security via S/MIME (libsmaug)

- libsmaug leverages DANE for object security in applications
  - Email is just *one* use of S/MIME
- libsmaug optionally uses full featured resolver

**Implementation**
- Open source
- C/C++
- [https://github.com/verisign/smaug](https://github.com/verisign/smaug) and [https://github.com/verisign/smaug-tbird-plugin](https://github.com/verisign/smaug-tbird-plugin)
Thunderbird Add-on
DANE for SMTP

- Connections between SMTP servers today can use TLS encryption opportunistically
- Even when encryption is used, it is vulnerable to attack:
  - Attackers can strip away the TLS capability
  - TLS certificates are often unauthenticated (self signed certificates)

- DANE can address both these vulnerabilities
  - Authenticate the certificate using a DNSSEC signed TLSA record
  - Use the presence of the TLSA record as an indicator that encryption must be performed (prevent downgrade)
DANE for SMTP

- SMTP over TLS, or SMTP + STARTTLS

- DANE can authenticate TLS for the SMTP connection between the mail server and the user’s mail client (MUA)

- DANE can authenticate TLS connections between SMTP servers ("MTA”s or Mail Transfer Agents)
Example TLSA record (for SMTP)

```
_25._tcp.mx1.freebsd.org. 2389 IN TLSA 3 0 1 (5EC0508C3F337D18509F41BFF9D8AB07FED588A132FA12FA1E223BA6B9403ACB )

_25._tcp.mx1.freebsd.org. 2389 IN RRSIG TLSA 8 5 3600 (20141023072418 20141009105807 39939 freebsd.org.

116DEQ7oP2lbEcOeJyPkJ8tYiGz4CzuDiqiMbr4Mzp390UWdej3kdaZ4t+1BT0dO3/o0nz0pp3HFsdDu+gkwT6YHJg4C6mi3STPciCP1tjbFuW/dv41PkCUa7kKj/qwPrR60kQmyvcuUoYgUDPbNYbJNJXai+mFai5WqLS2MEP15ydUn8KympnjHS5mVLVGXW0e7tLY1afQz1VrIeYsGW8YztMDYUpCXjWiq+YpCFv7rZ7IejQR6ot1M35CDsfj68eu0EAjx+HlqaTgdGyi6cMB+GduFwqkULDPIgi6Fu/3xb+srJrzur89YpHga90Cnz6nXJgQ6cXvSImZWbKuw== )
```

This is a domain-issued certificate (usage 3), which can be authenticated without a trusted CA.
Large adopters of SMTP + DANE

- Roughly 400 .com domains
  - us-core.com
  - omc-mail.com
  - five-ten-sg.com

- Quite a few are large email systems in Germany.
  - posteo.de
  - mailbox.org
  - umbkw.de
  - bund.de
  - denic.de
  - freebsd.org
  - debian.org, debian.net
  - ietf.org
  - nlnetlabs.nl
  - nic.cz
  - nic.ch
  - torproject.org
SMTP servers that support DANE

- Postfix MTA (works today, version 2.11 onwards)
- Exim (currently under development)

Quick start for Postfix:

```
postconf -e "smtpd_use_tls = yes"
postconf -e "smtp_dns_support_level = dnssec"
postconf -e "smtp_tls_security_level = dane"
```
Jabber / IM servers

- XMPP (Jabber) has seen some uptake of DANE.
- To authenticate the c2s and/or s2s portion of the XMPP protocol
- List of XMPP servers with DANE TLSA records:
  - [https://xmpp.net/reports.php#dnssecdane](https://xmpp.net/reports.php#dnssecdane)

Example:

```
_xmpp-server._tcp.mail.de. 3600 IN SRV 10 20 5269 jabber.mail.de.

_5269._tcp.jabber.mail.de. 600 IN TLSA 3 1 1 (A0315F0CF61CAC787140833C2C608550476246DDA54122D66BB339D5 0FBB10E3) - Uses End Entity cert
- Covers just the SPKI
- Encodes a hash
```
OpenPGPKEY

- OPENPGPKEY record
- Used to publish an OpenPGP public keys in the DNS
- DNSSEC signature provides authentication
- Spec under development, but RR code already assigned
Example OPENPGPKEY record

\texttt{sha256(username)[0:28].\_openpgpkey.<domain>}

e.g. for \texttt{shuque@huque.com}

1\textsuperscript{st} label: \texttt{sha256-hash("shuque")] truncated to 28 octets = adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101

2\textsuperscript{nd} label: \texttt{"\_openpgpkey"}

Remaining labels: domain portion of email address: \texttt{huque.com}

Resulting record looks like this:

\texttt{adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101.\_openpgpkey.huque.com.} IN OPENPGPKEY <base64 encoding of the openpgp key>
The promise of DANE

- Providing security to data in motion and data at rest
- Secure resting data objects
  - SMIMEA / OPENPGPKEY can secure email, documents, etc.
- Securing that secure data while in flight
  - TLSA secures TLS sessions: HTTPS, inter-SMTP, etc.
So, where are we today

- DANE has one proposed standard
  - TLSA, RFC 6698
  - There is a growing toolset
  - Mainly operational in inter-SMTP mail security

- Draft standards for email encryption and signing
  - SMIMEA
  - There is an open source library (libsmaug), a pilot MUA support in Thunderbird, and a DNS zone management portal

- OPENPGPKEY
  - There is an open source toolkits (libsmaug and.hashslinger)
Coordinated efforts include…

- National Cybersecurity Center of Excellence (NCCoE) just announced a building block: DNS-Based Secure Email
  - NCCoE is a public-private collaborative FFRDC focused on the implementation and practice of Internet security
  - Vendors work together within NCCoE to “build modular end-to-end reference designs”
  - Call for comments through August 14
  - Call for interest to become a vendor partner (US and non-US)
  - See: https://nccoe.nist.gov/dnssecureemail

- The Internet Society (ISOC) has a deployment program called Deploy 360
  - http://www.internetsociety.org/deploy360/resources/dane/
A glimpse into the future

- Imagine a future where you can send anyone encrypted email, and they can verify it came from you

- Imagine a future where connected to web servers can be encrypted, *and* we don’t have WebPKI vulnerabilities

- In the future, DANE will give us (the users) true end-to-end security