DANE/DNSSEC/TLS Testing in the Go6lab

Jan Žorž, ISOC/Go6 Institute, Slovenia
jan@go6.si
zorz@isoc.org

Acknowledgement

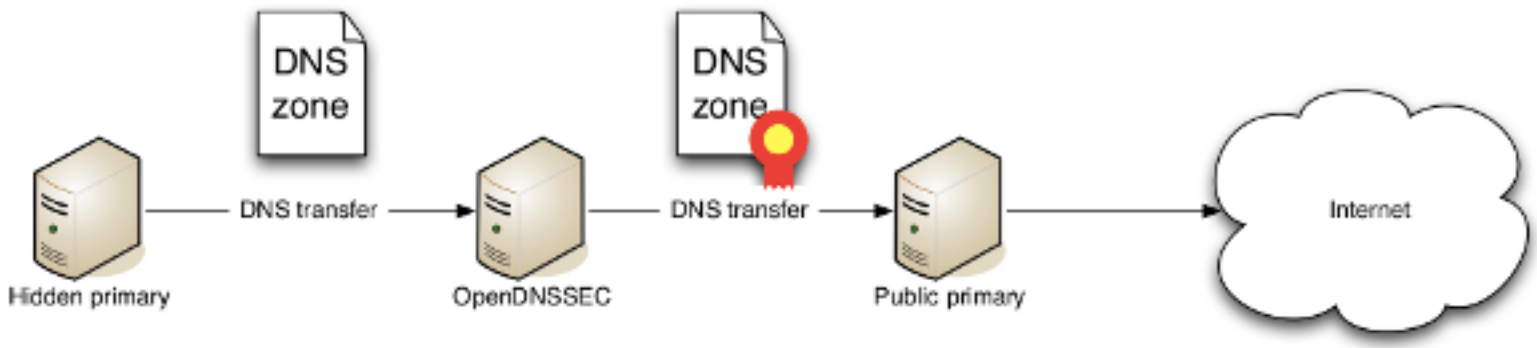
I would like to thank Internet Society to let me spend some of my ISOC working time in go6lab and test all this new and exciting protocols and mechanisms that makes Internet a bit better and more secure place…

DNSSEC implementation in go6lab

- Powerdns server (used as primary for non-signed domains) as “hidden” primary DNS server
- OpenDNSSEC platform for signing domains
- BIND9 DNS servers as secondaries to OpenDNSSEC to serve signed zones
- Virtualization used: PROXMOX 3.4
- OS templates: fedora-20, Centos6/7

DNSSEC implementation in go6lab

- "Bump in a wire"
- Two public "primary" servers
- Concept:



DNSSEC in go6lab

- That was fairly easy and it works very well.
- Implementation document used from Matthijs Mekking:

http://go6.si/docs/opendnssec-start-guide-draft.pdf

DANE experiment

- When DNSSEC was set up and functioning we started to experiment with DANE (DNS Authenticated Name Entities).
- Requirements:
  - DNSSEC signed domains
  - Postfix server with TLS support > 2.11
- We decided on Postfix 3.0.1

DANE

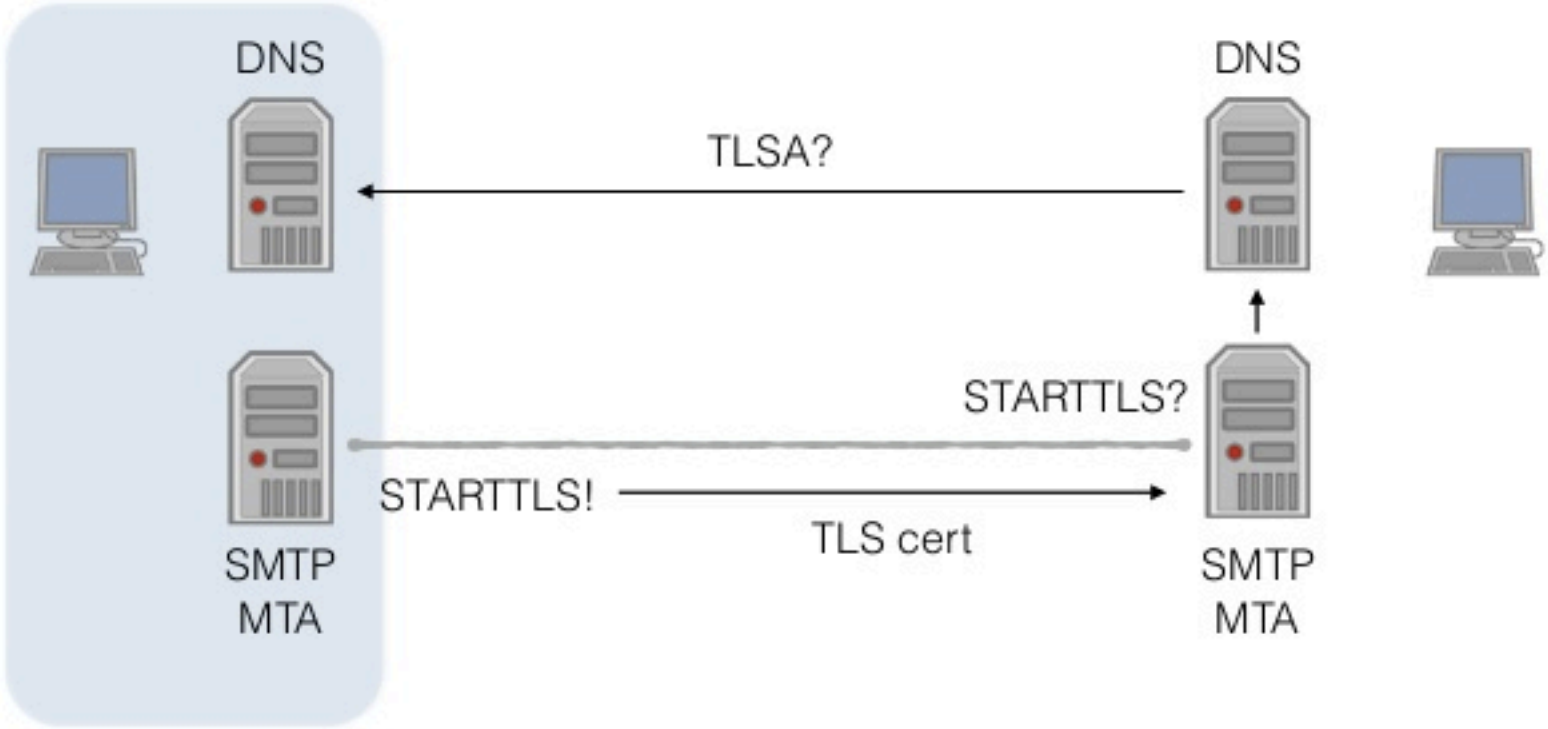
- TLSA record for mx.go6lab.si

_25._tcp.mx.go6lab.si. IN TLSA 3 0 1 B4B7A46F9F0DFEA0151C2E07A5AD7908F4C8B0050E7CC25908DA05E2 A84748ED

It's basically a hash of TLS certificate on mx.go6lab.si

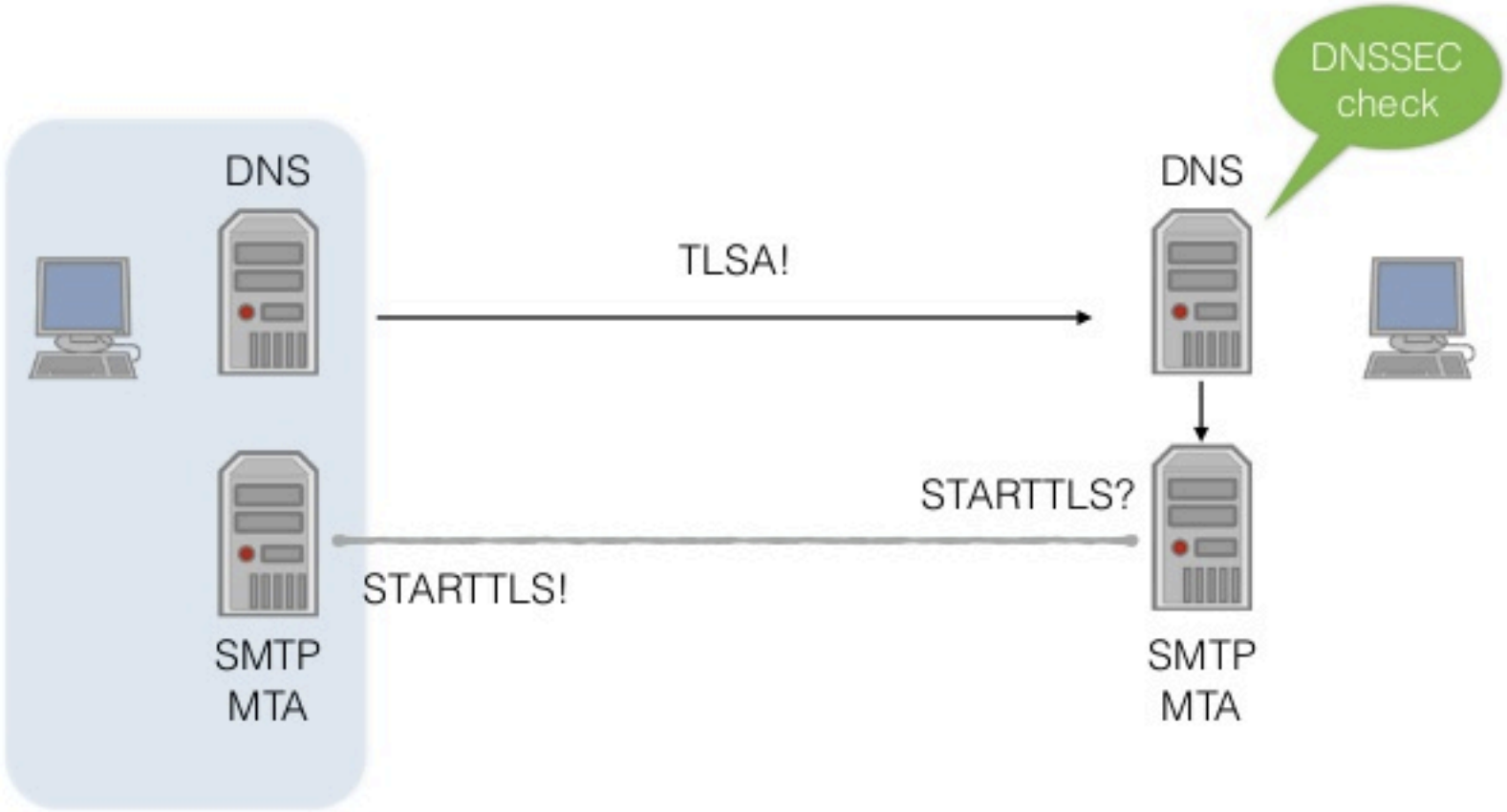
More about DANE:
http://www.internetsociety.org/deploy360/resources/dane/

TLS and SMTP



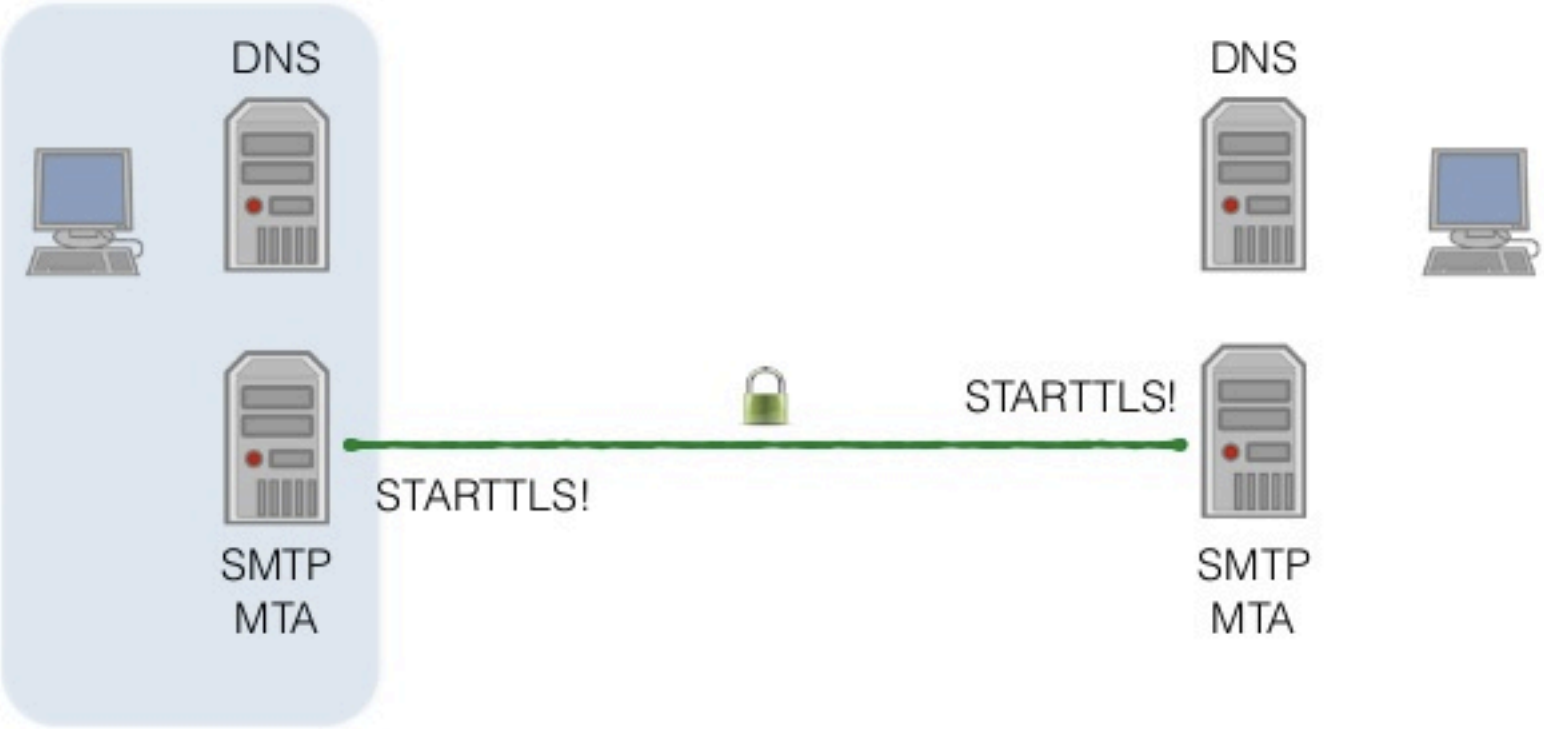
© Men & Mice http://menandmice.com

TLS and SMTP



© Men & Mice http://menandmice.com

TLS and SMTP



© Men & Mice http://menandmice.com

DANE verification

- Mx.go6lab.si was able to verify TLS cert to T-2 mail server and nlnet-labs and some others…

mx postfix/smtp[31332]: Verified TLS connection established to smtp-good-in-2.t-2.si[2a01:260:1:4::24]:25: TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits)

dicht postfix/smtp[29540]: Verified TLS connection established to mx.go6lab.si[2001:67c:27e4::23]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)

Postfix config

```
smtpd_use_tls = yes
smtpd_tls_security_level = may
smtpd_tls_key_file = /etc/postfix/ssl/server.pem
smtpd_tls_cert_file = /etc/postfix/ssl/server.pem
smtpd_tls_auth_only = no
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtp_tls_security_level = dane
smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtp_tls_loglevel = 1
tls_random_exchange_name = /var/run/prng_exch
tls_random_source = dev:/dev/urandom
tls_smtp_use_tls = yes
```

Malformed TLSA record

- We created a TLSA record with a bad hash (one character changed)
- Postfix failed to verify it and refused to send a message

mx postfix/smtp[1765]: Untrusted TLS connection established to mail-bad.go6lab.si[2001:67c:27e4::beee]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)

mx postfix/smtp[1765]: 3A4BE8EE5C: Server certificate not trusted

1M top Alexa domains and DANE

- We fetched top 1 million Alexa domains and created a script that sent an email to each of them (test-dnssec-dane@[domain])
- After some tweaking of the script we got some good results
- Then we built a script that parsed mail log file and here are the results:

Results

- Out of 1 million domains, 992,232 of them had MX record and mail server.
- Nearly 70% (687,897) of all attempted SMTP sessions to Alexa top 1 million domains MX records were encrypted with TLS
- Majority of TLS connections (60%) were established with trusted certificate
- 1,382 connections where remote mail server announced TLS capability failed with "Cannot start TLS: handshake failure"

More results

TLS established connections ratios are:

Anonymous: 109.753

Untrusted: 167.063

Trusted: 410.953

Verified: 128

Quick guide: Anonymous (opportunistic TLS with no signature), Untrusted (peer certificate not signed by trusted CA), Trusted (peer certificate signed by trusted CA) and Verified (verified with TLSA by DANE).

DANE Verified

Verified: 128 !!!

Mail distribution

Mail Servers	# Domains Handled	TLS State
google.com	125,422	Trusted
secureserver.net	35,759	Some Trusted, some no TLS at all
qq.com	11,254	No TLS
Yandex.ru	9,268	Trusted
Ovh.net	8.531	Most Trusted, with redirect servers having no TLS at all

Mail distribution

Mail Servers	# Domains Handled	TLS State
Emailsrvr.com	8,262	Trusted
Zohomail.com	2.981	Trusted
Lolipop.jp	1.685	No TLS
Kundenserver.de	2,834	Trusted
Gandi.net	2,200	Anonymous

DNSSEC? DANE?

None of these “big” mail servers (and their domains) are DNSSEC signed (that meant no DANE for them possible up to January 2016).

When do DANE things fail?

- Of course, with wrong certificate hash in TLSA record (refuses to send mail)
- If domain where MX record resides is not DNSSEC signed (can't trust the data in MX, so no verification)
- If TLSA record published in non-DNSSEC zone (can't trust the data in TLSA, so no verification)

When do things fail? (example)

- go6lab.si zone is signed, so is mx.go6lab.si
- there is TLSA for mx.go6lab.si, also signed
- Domain signed.si is signed and MX points to mx.go6lab.si
- Domain not-signed.si is not signed and MX points to mx.go6lab.si
- We send email to jan@signed.si and jan@not-signed.si (signed.si and not-signed.si are used just as examples)

When do things fail? (example)

When I send email to jan@signed.si (signed domain):

Verified TLS connection established to mx.go6lab.si[2001:67c:27e4::23]:25:

When I send email to jan@not-signed.si (not signed domain):

Anonymous TLS connection established to mx.go6lab.si[2001:67c:27e4::23]:25:

When do DANE verification also fail?

- Let's try to point MX record from signed domain to A/AAAA record in not-signed domain with TLSA that is also not signed (obviously) – mail.not-signed.si

Send mail to jan@signed.si when MX for signed.si points to mail.not-signed.si – DANE verification is not even started as chain of trust is broken

Postfix latest improvements ☺

postfix-3.1-20160103/HISTORY:

20160103

Feature: enable DANE policies when an MX host has a secure TLSA DNS record, even if the MX DNS record was obtained with insecure lookups. The existence of a secure TLSA record implies that the host wants to talk TLS and not plaintext. This behavior is controlled with smtp_tls_dane_insecure_mx_policy (default: "dane", other settings: "encrypt" and "may"; the latter is backwards-compatible with earlier Postfix releases). Viktor Dukhovni.

Let's Encrypt, DANE and mail

- Let's Encrypt recommends using '2 1 1' and '3 1 1' records
- Validity of LE cert is 90 days
- By default the underlying key is changed when renewing
- …so also cert hash is changed
- So, lot's of work if you plan to publish 3 1 1 TLSA
- using the '2 1 1' method leads to another issue – namely lack of an DST Root CA X3 certificate in the fullchain.pem file provided by the Let's Encrypt client
- So we need to fetch the DST Root CA X3 certificate and add it to fullchain.pem file and verify that it did not change from previous time we renewed…

Script to add DST Root CA X3

```
lynx --source https://www.identrust.com/certificates/trustid/root-download-x3.html | grep -v "\/textarea" | awk '/textarea/{x=NR+18;next}(NR<=x){print}' | sed -e '1i-----BEGIN CERTIFICATE-----\' | sed -e '$a-----END CERTIFICATE-----\' >> /etc/letsencrypt/live/mx.go6lab.si/fullchain.pem
```

Valid 3 1 1 and 2 1 1 TLSA records

10 mx.go6lab.si

DNSSEC ✓ TLSA ✓ SMTP ✓ Show Details

IP Addresses

91.239.96.23

2001:67c:27e4:0:0:0:0:23

Usable TLSA Records

2, 1, 1 563b3caf8cfef34c[...]59784830df9e5b2b

3, 1, 1 c8ac0381d0b8c901[...]51001406bfca0db5

But…

- At next certificate renew, by default underlying key will change and 3 1 1 TLSA record will become invalid…
- Labor wise, we need to keep the underlying key through the renewals
- --csr option in letsencrypt-auto client
- In direcotry “examples” there is “generate-csr.sh” file

Stable underlying key…

```
./generate-csr.sh mx.go6lab.si
Generating a 2048 bit RSA private key
................+++
..+++
writing new private key to 'key.pem'
-----
You can now run: letsencrypt auth --csr csr.der
```

Renewals and hashes…

- Now we are using the same underlying key for automatic renewals of certificate, so hash does not change and 3 1 1 TLSA record works.
- We'll rotate the underlying key when we decide to and being driven by human intervention (and also change the TLSA).
- ./letsencrypt-auto certonly -t --debug --renew -a standalone --csr ./mx.go6lab.si.der –keep
- Of course, we add DST Root CA X3 certificate to fullchain.pem

go6.si

DNSSEC ✓ TLSA ✓ SMTP ✓

The domain lists the following MX entries:

10 mx.go6lab.si

DNSSEC ✓ TLSA ✓ SMTP ✓ Show Details

IP Addresses

91.239.96.23

2001:67c:27e4:0:0:0:0:23

Usable TLSA Records

```
3, 1, 1 89bbad366051875b[...]ef9a0b63ed166c25
```

```
2, 1, 1 563b3caf8cfef34c[...]59784830df9e5b2b
```

More reading:

http://www.internetsociety.org/deploy360/blog/2016/01/lets-encrypt-certificates-for-mail-servers-and-dane-part-1-of-2/

http://www.internetsociety.org/deploy360/blog/2016/03/lets-encrypt-certificates-for-mail-servers-and-dane-part-2-of-2/

Conclusions

- 70% of email can be encrypted in some way, you just need to enable TLS on your server
- Low number of DNSSEC signed domains/ servers
- Even lower number of DANE/TLSA verified servers/connections
- It's easy, go and do it – it's not the end of the world and it helps with verifying who are you sending emails to – and vice versa ;)

Conclusions II.

- DANE verification failed (or was aborted) if DNSSEC chain of trust is not fully established and complete along the whole way.
- TLSA in not-signed DNS zones would not help you much preventing your correspondents sending emails to server-in-the-middle (if you are not running latest bleeding edge development version of Postfix)
- DNSSEC/DANE is easy, but please understand what are you doing before implementing it in production…

Q&A

Questions? Protests? Suggestions? Complaints?

jan@go6.si
zorz@isoc.org