Hamza Boulakhrif  Guido Kroon

Supervisor:
Michiel Leenaars (NLnet Foundation)

hamza.boulakhrif@os3.nl, guido.kroon@os3.nl

UNIVERSITY OF AMSTERDAM

Faculty of Physics, Mathematics and Informatics
Graduate School of Informatics
System and Network Engineering MSc

February 6, 2015
Introduction

- Classic CA model
  - Trusted Certificate Authorities
  - Pre-configured CA certificate collections
- DANE
  - DNSSEC chain of trust
  - TLSA RRs
  - PKIX validation (optional)
Classic CA model

Figure 1: Classic validation.
Figure 2: DANE validation.
Basic TLSA RR

- Basically a customised SRV RR
- Service, Proto, Name, Class fields
- Certificate Usage
- Selector
- Matching Type
- Certificate Association Data

**TLSA RR format**
<Service)._Proto._Name Class TLSA Usage Selector Mtype Data

**TLSA RR example**
_443._tcp.dane.internet.nl. IN TLSA (  
  0 0 1 d2abde240d7cd3ee6b4b28c54df034b9  
  7983a1d16e8a410e4561cb106618e971 )
The four different Certificate Usages of DANE.

- **Usage 1 (Server Certificate Constraint)**
  TLSA RR specifies which EE certificate should be used for the domain.

- **Usage 3 (Domain-issued Certificate)**
  TLSA RR specifies the TLS certificate that should be used for the domain, without PKIX validation.
Usage 0 (CA Constraint)
TLSA RR specifies which CA will provide TLS certificates for the domain.

Usage 2 (Trust Anchor Assertion)
TLSA RR specifies which trust anchor will provide TLS certificates for the domain, allowing the use of a CA not included in the CA certificate collection of the application.
Can a test suite be devised to allow developers and implementers to validate the reliability and consistency of an implementation of DANE, and its ability to correctly handle unforeseen input or deviations from the official TLSA syntax as per RFC 6698?
The scope for this research.

- Analysis of RFC6698
- Extensible test suite
  - Usages
- Test DANE implementations

Not part of scope research:

- (Re)writing DANE-tools
- (Re)compiling of DANE-tools
Approach

The approach for this research.

- Analysis of DANE RFC 6698 (and RFC 6394)
- Deployment of environment
- Build test suite in environment
- Test DANE implementations
The test suite is built by using:

- BIND
- Apache
Experiments (1)

- GnuTLS
- ldns-dane
- DNSSEC/TLSA Validator (browser add-on)

Figure 4: GNUTLS Danetool
Experiments (2)

Test cases that are devised by the analysis of the DANE specification.

- (Non-)existing usages
- (Non-)existing Selectors
- (Non-)existing Matching types
- Combination of Selector and Matching type incorrect
- (In)correct hash (type)
- Expired certificates
- Unsigned DNSSEC chain
- Wildcard usage
- Incorrect signed certificates
Results

- GnuTLS
  - No PKIX validation (intentional).
- ldns-dane
  - Specify CA certificates manually for PKIX validation.
- DNSSEC/TLSA Validator
  - No PKIX validation, even though it claims to.

Figure 5: DNSSEC/TLSA Validator without proper PKIX validation.
Conclusion

Based on the results, a couple of conclusions can be derived.

- RFC 6698
  - Interpretation

- Test suite
  - Good
  - Bad
  - Grey

- BIND
  - Test cases
  - Limitations
Some noteworthy details, which lie outside of the scope of this project:

- Think of more test cases
  - Proxy in front of BIND
- Test cases for all usages (CA Contraint)
- Source code analysis of DANE implementations
- Complete DANE support in DANE implementations
The End