DNSSEC serialization for TLS

Paul Hoffman
VPN Consortium

Overview

- Why we might care
- What serialization looks like
- How to get this data into TLS
- How this work might proceed in the IETF

Important note: I am not Adam Langley, nor Dan Kaminsky

Why DNSSEC serialization is interesting for TLS clients

- Getting all the DNSSEC data needed to validate a particular DNS record with no DNS lookups, as one blob, is much faster than getting it using recursive DNS requests
- Some clients can’t reliably get DNSSEC data (filtering firewalls, broken proxies, ...)
- Doing DNSSEC validation in the client means that the client doesn’t need to trust a DNS resolver to validate DNS

What serialized DNSSEC looks like

- A binary blob containing all the data that would be needed by a validating DNSSEC resolver in one step
- Some optimizations are possible if you want to save ~20% of the size (less than 1K)
- There have been a few proposals for what the structure of the blob should be

Getting this data into TLS

- New TLS extension
- New PKIX extension in an end-entity certificate
- New PKIX extension in a superfluous certificate
- New type of CertificateStatusRequest
- New OCSP extension carried in a OCSPStatusRequest
- Not covered: do another request (DNS, HTTP, ...)

New TLS extension

- + Easy to define new TLS extensions
- + Single purpose extension that does not mix semantics with other extensions
- - Some TLS servers can’t be extended easily

New PKIX extension in the EE cert

- + Easy to define new PKIX extensions
- - Only works for self-issued certs because the cert needs to be re-issued very often (possibly about once an hour)
- + Self-issued EE certs are easy to re-issue

New PKIX extension in a superfluous cert

- + Easy to define new PKIX extensions
- + Superfluous certs are accepted by all current browsers
- - Superfluous certs are illegal in the TLS spec

New type of CertificateStatusRequest

- + Easy to define new CertificateStatusRequest types
- - Unclear how easy it is for clients and servers to be extended this way
- - Doesn’t fall under the semantics of a “certificate status request” so this might not be allowed

New OCSP extension carried in a OCSPStatusRequest

- + Easy to define new OCSP extensions
- + Most browsers already accept OCSP
- - Not clear how much effort is needed for a browser to reach into an OCSP message to grab the extension

How this might proceed in the IETF

- Need to standardize on a serialization without bikeshedding, premature optimization, and so on
- Need to decide which mechanism will be used to carry this in TLS
- These two should be able to proceed in parallel