Delegated Credentials

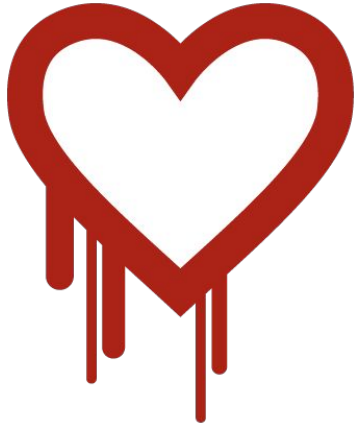
TLS WG

IETF 102

https://tools.ietf.org/html/draft-ietf-tls-subcerts-01

R. Barnes, S. Iyengar, N. Sullivan, E. Rescorla

Background



Motivation

Reduce the exposure of certificate private keys to memory disclosure vulnerabilities and compromise of TLS termination infrastructure without compromising performance.

Adopted

October 2017

Latest update

July 2018 - mostly editorial

Details

Construction

Empty “Delegated Credentials” TLS extension send from client

If accepted, the server

- Sends DC in TLS extension response
- Uses the DC private key to create CertificateVerify (instead of certificate key)

Client validates DC was correctly signed by EE certificate and date is valid.

Requires new OID in certificate.

Details

Structure

```
struct {
  uint32 valid_time;
  opaque public_key<0..2^16-1>;
} Credential;

struct {
  Credential cred;
  SignatureScheme scheme;
  opaque signature<0..2^16-1>;
} DelegatedCredential;
```

The signature of the DelegatedCredential is computed over the concatenation of:

1. 0x20 repeated 64 times.
2. "TLS, server delegated credentials"
3. A single 0 byte
4. Big endian serialized 2 bytes ProtocolVersion of the TLS version
5. DER encoded X.509 certificate used to sign the DelegatedCredential.
6. Big endian serialized 2 byte SignatureScheme scheme
7. The Credential structure

Implementation status

Go implementation of the current draft (tls-tris).

BoringSSL (bssl) implementation nearly complete

Interoperability between a bssl client and a tris server.

We expect to have comprehensive interop testing done soon.

Cloudflare has a plan for serving DCs on behalf of its customers, targeting Fall 2018.

Question #1: Which OID to use

Current draft: *new “id-ce-delegationUsage” OID*

Working OID: for delegationUsage X.509 extension:1.3.6.1.4.1.44363.44

Should we switch to the reserved OID for IETF Security OID?

Should we consider changing this to an ExtendedKeyUsage?

Proposal #2: Introduce a TLS Feature extension

Current draft defines an OID

Proposal: Add optional TLS Feature enum value for Must-Use-DC

Serving a DC becomes required for a DC-capable certificate

This reduces risk of cross-protocol attacks and signing oracles

Proposal #3: Bind DC to the handshake signature scheme

Currently, a delegated credential can be used for any signature_scheme that its key type is capable of doing

- e.g. rsa_pss_pss_sha256 *and* rsa_pss_pss_sha384

The proposal (https://github.com/tlswg/tls-subcerts/pull/7) is to also include a signature_scheme in the binding of the delegated credential to the EE certificate.

Pros: Tighter control for DC issuer of how DC is to be used

Cons: More DCs to mint in some circumstances

Proposal #4: Drop support for TLS 1.2

Stacks that will be updated to use DC will likely already have TLS 1.3.
Retrofitting this to TLS 1.2 introduces a lot of complexity.

Pros: Tighter control for DC issuer of how DC is to be used

Cons: More DCs to mint in some circumstances

Nice-to-have for last call

Formal verification

Any takers?

Delegated Credentials

TLS WG

IETF 102

https://tools.ietf.org/html/draft-ietf-tls-subcerts-01

R. Barnes, S. Iyengar, N. Sullivan, E. Rescorla