Connection ID

draft-rescorla-tls-connection-id-02

Eric Rescorla
Mozilla
ekr@rtfm.com

Hannes Tschofenig
Arm Limited
hannes.tschofenig@arm.com

Thomas Fossati
Nokia
thomas.fossati@nokia.com

Tobias Gondrom
Huawei
tobias.gondrom@gondrom.org

Recap from last time

- Lack of Connection IDs clearly a problem for NATs/IoT, etc.
- Connection IDs are also a clear privacy problem
  - Lots of proposals for how to do privacy preserving Conn IDs
  - ... but they're complicated and none of them seem totally baked
- Proposal: use a fixed connection ID for now
  - In an extension
  - We can always replace it later
- This got pulled out of DTLS and into its own draft

Basic idea

- IDs are used if client offers and server answers
  - On all (non-0RTT)? encrypted records
- Each side *sends* with the other's ID
  - Because IDs are unframed, 0-length IDs are just omitted

DTLS 1.2

```
struct {
   ContentType type;
   ProtocolVersion version;
   uint16 epoch;
   uint48 sequence_number;
   opaque cid[cid_length];               // New field
   uint16 length;
   select (CipherSpec.cipher_type) {
      case block:  GenericBlockCipher;
      case aead:   GenericAEADCipher;
   } fragment;
} DTLSCiphertext;
```

DTLS 1.3[*]

```
struct {
    ContentType opaque_type = 23; /* application_data */
    uint32 epoch_and_sequence;
    opaque cid[cid_length];               // New field
    uint16 length;
    opaque encrypted_record[length];
} DTLSCiphertext;

struct {
  uint16 short_epoch_and_sequence;  // 001ESSSS SSSSSSSS
  opaque cid[cid_length];               // New field
  opaque encrypted_record[remainder_of_datagram];
} DTLSShortCiphertext;
```

[*] Not in the draft. Ugh.

Connection ID Update (TLS 1.3 only)

```
enum {
    cid_immediate(0), cid_spare(1), (255)
} ConnectionIdUsage;

struct {
    opaque cid<0..2^8-1>;
    ConnectionIdUsage usage;
} NewConnectionId;
```

- `cid_immediate` means “delete all your older conn ids”
- `cid_spare` means “add to the valid conn ids”
- I am not sure this is ideal

Open Issues

- Do we need a way to tell if a CID is present
  - to deal with servers which have both CID and non-CID connections
- Do we need CID update for TLS 1.2 (how?)
- The record sequence number allows cross-CID linkage
  - Solution: adopt the technique we used for QUIC of predictable jumps

Other issues? WG adoption?

Options for TLS 1.2 Post-Handshake CID Refresh

- Do nothing
- TLS 1.2 renegotiation
- Port over TLS 1.3 post-handshake messaging
  - I think we'd need to deprecate renegotiation