Ticket-Based Identity Pinning

draft-sheffer-tls-pinning-ticket

Yaron Sheffer, Daniel Migault

IETF 97, Seoul

The Problem

- The problem we're solving: **misissued (fake) certificates**
- The solution approach: **Trust On First Use** (TOFU)

- Certificate pinning is standardized: HPKP (RFC 7469), but…
  - The standard is not implemented
  - People are still doing their own stuff, with occasional fails
  - Also, it's only good for HTTP

Solution Overview

- Instead of certificate pinning, we suggest **identity pinning**
- Each client gets a unique shared secret with the server (cluster), ensuring continuity of the server's authentication
- We use client-side tickets to store this shared secret in a scalable way
- All at the TLS level, specified for TLS 1.3
  - Also, implemented as a fork of Mint

Initial Connection

- Client requests a ticket
- Server generates a ticket and sends it back
  - In the Encrypted Extensions
- The ticket is opaque to the client, contains a value generated from the TLS handshake, and is protected
  - The client can compute the value independently
- The ticket is time limited
  - This is a commitment by the server!
- The client stores the ticket and the corresponding secret value for the ticket’s lifetime

Further Connections

- The client sends the ticket
- If this is the correct server:
  - It decrypts the ticket
  - And sends back a proof of its knowledge of the secret value
- Otherwise, the client knows there's something wrong
- The client must still authenticate the server using the server's cert: this is a **second factor** server auth

Additional Considerations

- The ticket is protected with a key that may be rotated regularly
  - No manual intervention, unlike HPKP
- In a cluster, a shared key between all members
  - Similar to normal session resumption
- The server certificate may be replaced or even revoked while there are outstanding tickets
- We included a “ramp down” mode, ensuring the server is not bricked if you decide to stop using this protocol

Thank You!