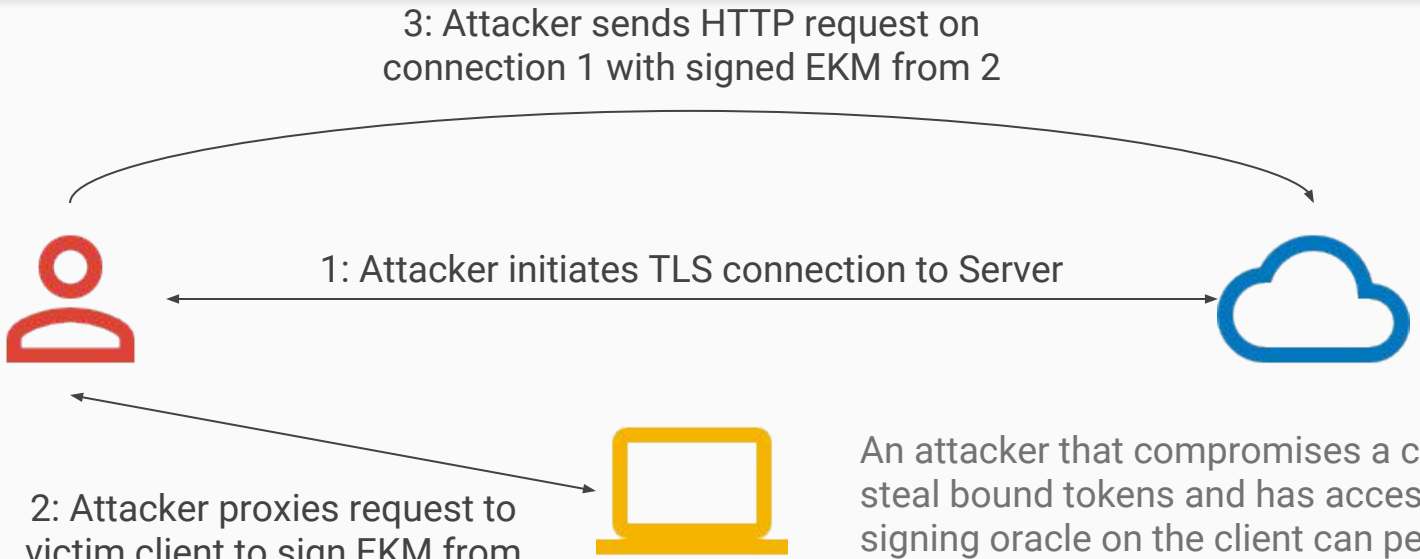
Token Binding for 0-RTT TLS 1.3 Connections

draft-ietf-tokbind-tls13-0rtt
Nick Harper
IETF 98

What’s Changed

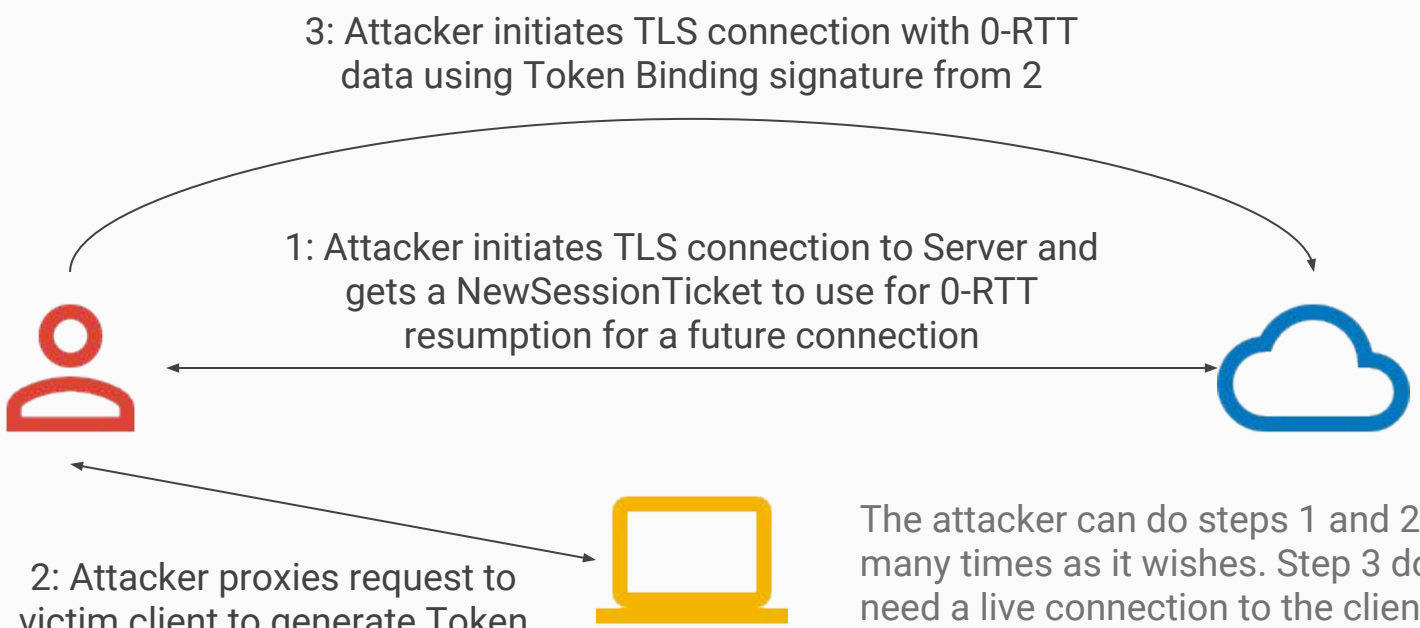
- New language in Security Considerations for Proof of Possession
- Client switches from 0-RTT exporter to normal exporter during connection
- Client indicates 0-RTT exporter is in use with extension in TokenBinding struct

Proof of Possession



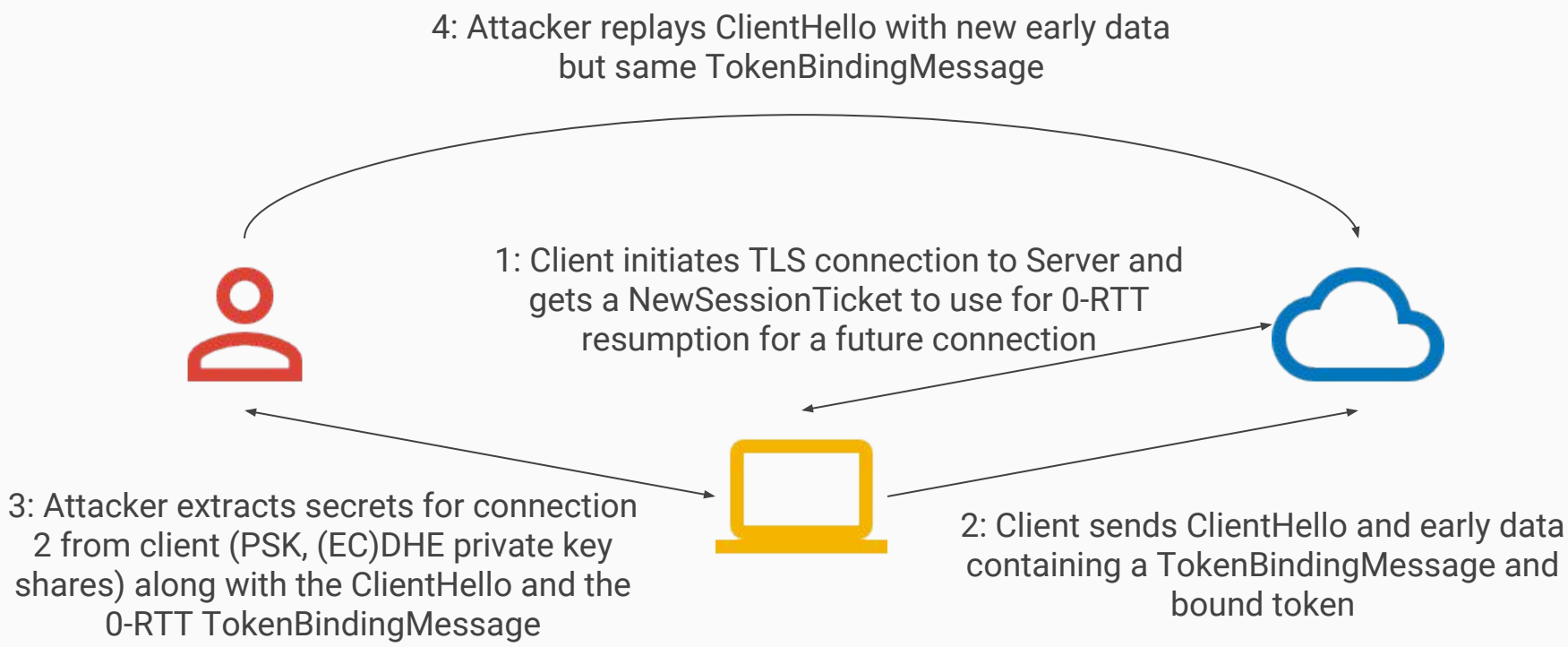
An attacker that compromises a client to steal bound tokens and has access to a signing oracle on the client can perform this attack to replay bound tokens on a new connection so long as the attacker maintains the connection with the client.

Once the attacker loses the connection to the client, the attacker cannot do this on a new connection.



The attacker can do steps 1 and 2 as many times as it wishes. Step 3 does not need a live connection to the client.

After the attacker's access to the client machine has been severed, the attacker can still send valid 0-RTT Token Binding signatures for up to the lifetime of a NewSessionTicket (up to 7 days).

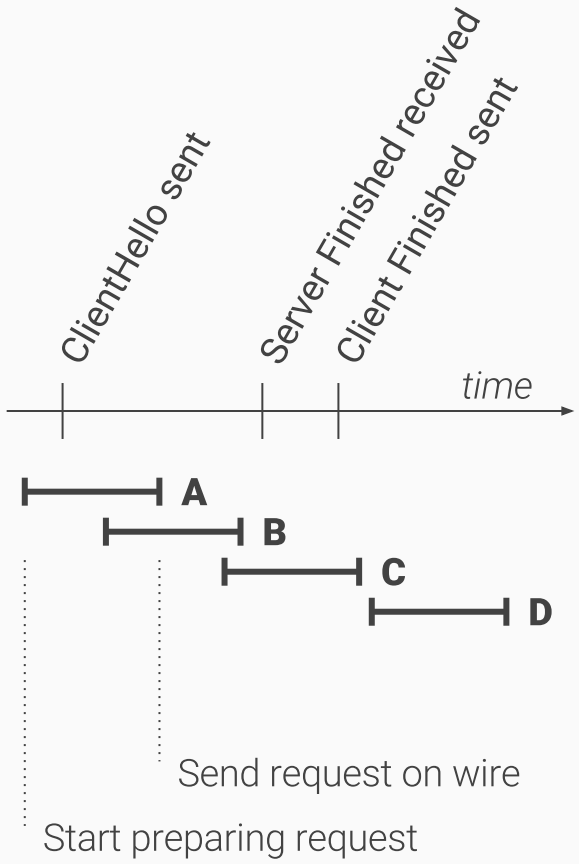


This attack scenario never has the attacker use the private key, but they do use other secrets from the client. The maximum time window for carrying out such an attack after being removed from the client is dependent on how long the server will accept the replayed ClientHello.

Changing Exporters

Client switches from 0-RTT exporter to normal exporter as soon as possible

Client can still send request with 0-RTT exporter after sending Finished, for the case when the client starts preparing the request before the normal exporter is available, but sends it after the handshake is complete (e.g. request C to the right).



An argument for always using the 0-RTT exporter

A server that accepts TB and 0-RTT on the same connection means that the server considers the security properties of the 0-RTT exporter sufficient for at least some requests.

On those requests, it is logical for the server to accept them with 0-RTT exporter post-handshake as well.

A server cannot reject early data based on its contents, so the server's decision to accept the 0-RTT exporter for some requests must apply to all requests.

Replay Protection TLS Extension

- Server sends it to indicate to client that the server implements some sort of replay protection of 0-RTT Token Binding signatures
- Replay of 0-RTT Token Binding signatures (with new application data) is only possible if the attacker has the PSK for that connection
- Such an attacker could also generate new Token Binding signature for a new PSK instead of replaying one

Does this proposed extension actually provide any value?

Upcoming changes

- Fix language in 2.1.1 around switching exporters to say:
  - “All requests which the client starts processing to send after the client sends its Finished message MUST use the exporter_secret for their token bindings.”
- Section 2.2.1:
  - Clarify that a change in Token Binding key parameter that causes a server to reject early data also includes whether or not Token Binding was negotiated.