Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 2, 2019

M. Boucadair
Orange
T. Reddy
McAfee

January 29, 2019

Using Early Data in DOTS
draft-boucadair-dots-earlydata-00

Abstract

This document discusses to what extent it is safe to send DOTS signal channel messages as Early Data in TLS 1.3.

This document is not intended to be published as an RFC. It is edited to help understanding the conclusion about the safeness of using DOTS signal channel messages as early data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 2, 2019.

Copyright Notice

Table of Contents

1. Context

Section E.5 of [RFC8446] states the following:

> Replayable 0-RTT data presents a number of security threats to TLS- using applications, unless those applications are specifically engineered to be safe under replay (minimally, this means idempotent, but in many cases may also require other stronger conditions, such as constant-time response).
>
> ...
>
> Application protocols MUST NOT use 0-RTT data without a profile that defines its use. That profile needs to identify which messages or interactions are safe to use with 0-RTT and how to handle the situation when the server rejects 0-RTT and falls back to 1-RTT.

To that aim, [I-D.ietf-dots-signal-channel] includes the following:

> Section 8 of [RFC8446] discusses some mechanisms to implement to limit the impact of replay attacks on 0-RTT data. If the DOTS server accepts 0-RTT, it MUST implement one of these mechanisms. A DOTS server can reject 0-RTT by sending a TLS HelloRetryRequest. The DOTS signal channel messages sent as early data by the DOTS client are idempotent requests. As a reminder, Message ID

(Section 3 of [RFC7252]) is changed each time a new CoAP request is sent, and the Token (Section 5.3.1 of [RFC7252]) is randomized in each CoAP request. The DOTS server(s) can use Message ID and Token in the DOTS signal channel message to detect replay of early data, and accept 0-RTT data at most once. Furthermore, 'mid'

value is monotonically increased by the DOTS client for each mitigation request, attackers replaying mitigation requests with lower numeric 'mid' values and overlapping scopes with mitigation requests having higher numeric 'mid' values will be rejected systematically by the DOTS server.

Owing to the aforementioned protections, especially those afforded by CoAP deduplication (Section 4.5 of [RFC7252]) and RFC 8446 anti-replay mechanisms, all DOTS signal channel requests are safe to transmit in TLS 1.3 as early data.

This document is intended to provide more elaboration to motivate the above conclusion included in [I-D.ietf-dots-signal-channel].

2. Reminder

DOTS signal channel relies on CoAP methods (GET, PUT, and DELETE) that are designed to adhere to the following design (Section 5.1 of [RFC7252]):

CoAP supports the basic methods of GET, POST, PUT, and DELETE, which are easily mapped to HTTP. They have the same properties of safe (only retrieval) and idempotent (you can invoke it multiple times with the same effects) as HTTP (see Section 9.1 of [RFC2616]). The GET method is safe; therefore, it MUST NOT take any other action on a resource other than retrieval. The GET, PUT, and DELETE methods MUST be performed in such a way that they are idempotent. POST is not idempotent, because its effect is determined by the origin server and dependent on the target resource; it usually results in a new resource being created or the target resource being updated.

Note also that Message ID (Section 3 of [RFC7252]) is changed each time a new CoAP request is sent, and the Token (Section 5.3.1 of [RFC7252]) is randomized in each CoAP request. Message ID is particularly used by a CoAP implementation for message deduplication

as discussed in Section 4.5 of [RFC7252].

3. Replay the Same Request to the Same DOTS Server

This attack assumes that an eavesdropper who can observe the 0-RTT data from a DOTS client and then replays the ClientHello and early data to the same DOTS server.

The DOTS server uses Message ID and Token in the DOTS signal channel message to detect replay of early data, and accepts 0-RTT data at most once.

4. Fork a Request to Multiple Servers

This attack assumes that an eavesdropper who can observe the 0-RTT data from a DOTS client to a DOTS server and then replays the ClientHello and early data to other DOTS servers.

Obviousily, the replayed message will be discarded if distinct credentials are used per DOTS server or if the scope of the request is not under the responsibility of a DOTS server.

As a reminder, the DOTS servers in the same domain have to maintain a globally consistant server state to handle the following scenarios:

- DOTS client using different DOTS servers for DOTS signal and data channel protocols, synchronization of server state is required to detect conflicts between mitigation requests and existing accept-lists.

- DOTS clients using different DOTS servers to send mitigation requests, synchronization of server state is essential to detect conflicting mitigation requests from DOTS clients.

- DOTS client sends mitigation requests with overlapping scopes to different DOTS servers, synchronization of server state is essential to detect conflicting mitigation request from the DOTS clients.

It is recommended to implement RFC 8446 anti-replay mechanisms by DOTS servers of a domain to accept 0-RTT data at most once and

silently discard the duplicate the request. Note that duplicate requests will also be discarded due to conflict detection policies described in [I-D.ietf-dots-signal-channel] (overlapping scopes).

As a side note, the procedure to select and/or contact DOTS servers when multiple servers are configured to a DOTS client is out of scope of [I-D.ietf-dots-signal-channel].

5. Fork a Request to Multiple Server-Domain Gateways

This attack assumes that an eavesdropper who can observe the 0-RTT data from a DOTS client to a server-domain DOTS gateway and then replays the ClientHello and early data to other server-domain DOTS gateways.

The ultimate DOTS server (i.e., the server to which the requests are relayed by the server-domain gateways) uses then Message ID and Token in the DOTS signal channel messages to detect replay of early data, and accepts 0-RTT data at most once.

6. Fork a Request to Multiple Client-Domain Gateways

This attack assumes that an eavesdropper who can observe the 0-RTT data from a DOTS client to a client-domain DOTS gateway and then replays the ClientHello and early data to other client-domain DOTS gateways.

If only one DOTS server is configured to all these client-domain gateways, then this DOTS server will detect duplicate requests because all these requests will expose the same Message ID, and Token.

If multiple DOTS servers are deployed, then the measures described in Section 4 have to be followed.

7. Block a Response from a DOTS Server or DOTS Gateway

This attack assumes the following:

- The DOTS client is provisioned with multiple DOTS servers (or DOTS gateways).

- The attacker blocks the response received from the DOTS server (or DOTS gateway) for early data.

- In the absence of the response, the DOTS client contacts another DOTS server (or DOTS gateway).

It is recommended to implement RFC 8446 anti-replay mechanisms by DOTS servers of a domain to accept 0-RTT data at most once and silently discard the duplicate the request.

Note that when the new request is received by another DOTS server, conflict detection discussed in [I-D.ietf-dots-signal-channel] will be used. The duplicate request will be rejected by the DOTS server because the mitigation request has overlapping target with a previous mitigation request from the same DOTS client.

8. Security Considerations

The document discusses security considerations related to the use of TLS 1.3 0-RTT feature for DOTS signal channel messages.

9. IANA Considerations

This document does not require any action from IANA.

10. Normative References

[I-D.ietf-dots-signal-channel]
K, R., Boucadair, M., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", draft-ietf-dots-signal-channel-27 (work in progress), January 2019.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol

Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
<https://www.rfc-editor.org/info/rfc8446>.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Tirumaleswar Reddy
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore, Karnataka 560071
India

Email: TirumaleswarReddy_Konda@McAfee.com