Workgroup: TLS
Internet-Draft: draft-ietf-tls-dtls-rrc-02
Updates: 6347 (if approved)
Published: 26 November 2021
Intended Status: Standards Track
Expires: 30 May 2022
Authors: H. Tschofenig, Ed. T. Fossati
Arm Limited Arm Limited

Return Routability Check for DTLS 1.2 and DTLS 1.3

Abstract

This document specifies a return routability check for use in context of the Connection ID (CID) construct for the Datagram Transport Layer Security (DTLS) protocol versions 1.2 and 1.3.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Transport Layer Security Working Group mailing list (tls@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tls/.

Source for this draft and an issue tracker can be found at https://github.com/tlswg/dtls-rrc.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In "classical" DTLS, selecting a security context of an incoming DTLS record is accomplished with the help of the 5-tuple, i.e. source IP address, source port, transport protocol, destination IP address, and destination port. Changes to this 5 tuple can happen for a variety reasons over the lifetime of the DTLS session. In the IoT context, NAT rebinding is common with sleepy devices. Other examples include end host mobility and multi-homing. Without CID, if the source IP address and/or source port changes during the lifetime of an ongoing DTLS session then the receiver will be unable to locate the correct security context. As a result, the DTLS handshake has to be re-run. Of course, it is not necessary to re-run the full handshake if session resumption is supported and negotiated.

A CID is an identifier carried in the record layer header of a DTLS datagram that gives the receiver additional information for selecting the appropriate security context. The CID mechanism has been specified in [I-D.ietf-tls-dtls-connection-id] for DTLS 1.2 and in [I-D.ietf-tls-dtls13] for DTLS 1.3.

Section 6 of [I-D.ietf-tls-dtls-connection-id] describes how the use of CID increases the attack surface by providing both on-path and off-path attackers an opportunity for (D)DoS. It then goes on describing the steps a DTLS principal must take when a record with a CID is received that has a source address (and/or port) different from the one currently associated with the DTLS connection. However, the actual mechanism for ensuring that the new peer address is willing to receive and process DTLS records is left open. This document standardizes a return routability check (RRC) as part of the DTLS protocol itself.

The return routability check is performed by the receiving peer before the CID-to-IP address/port binding is updated in that peer's session state database. This is done in order to provide more confidence to the receiving peer that the sending peer is reachable at the indicated address and port.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document assumes familiarity with the CID format and protocol defined for DTLS 1.2 [I-D.ietf-tls-dtls-connection-id] and for DTLS 1.3 [I-D.ietf-tls-dtls13]. The presentation language used in this document is described in Section 4 of [RFC8446].

3. RRC Extension

The use of RRC is negotiated via the rrc DTLS-only extension. On connecting, the client includes the rrc extension in its ClientHello if it wishes to use RRC. If the server is capable of meeting this requirement, it responds with a rrc extension in its ServerHello. The extension_type value for this extension is TBD1 and the extension_data field of this extension is empty. The client and server MUST NOT use RRC unless both sides have successfully exchanged rrc extensions.

Note that the RRC extension applies to both DTLS 1.2 and DTLS 1.3.

4. The Return Routability Check Message

When a record with CID is received that has the source address of the enclosing UDP datagram different from the one previously associated with that CID, the receiver MUST NOT update its view of the peer's IP address and port number with the source specified in the UDP datagram before cryptographically validating the enclosed record(s) but instead perform a return routability check.

```
enum {
    invalid(0),
    change_cipher_spec(20),
    alert(21),
    handshake(22),
    application_data(23),
    heartbeat(24),  /* RFC 6520 */
    return_routability_check(TBD2), /* NEW */
    (255)
} ContentType;

uint64 Cookie;

enum {
    path_challenge(0),
    path_response(1),
    reserved(2..255)
} rrc_msg_type;

struct {
    rrc_msg_type msg_type;
    select (return_routability_check.msg_type) {
        case path_challenge: Cookie;
        case path_response:  Cookie;
    };
} return_routability_check;
```

The newly introduced return_routability_check message contains a cookie. The cookie is a 8-byte field containing arbitrary data.

The return_routability_check message MUST be authenticated and encrypted using the currently active security context.

The receiver that observes the peer's address and or port update MUST stop sending any buffered application data (or limit the data sent to a TBD threshold) and initiate the return routability check that proceeds as follows:

1. A cookie is placed in a return_routability_check message of type path_challenge;

2. The message is sent to the observed new address and a timeout T is started;

3. The peer endpoint, after successfully verifying the received return_routability_check message echoes the cookie value in a return_routability_check message of type path_response;

4. When the initiator receives and verifies the return_routability_check message contains the sent cookie, it updates the peer address binding;

5. If T expires, or the address confirmation fails, the peer address binding is not updated.

After this point, any pending send operation is resumed to the bound peer address.

5. Example

The example TLS 1.3 handshake shown in Figure 1 shows a client and a server negotiating the support for CID and for the RRC extension.

```
       Client                                           Server

Key  ^ ClientHello
Exch | + key_share
     | + signature_algorithms
     | + rrc
     v + connection_id=empty
                                -------->
                                                 ServerHello  ^ Key
                                                +  key_share  | Exch
                                         + connection_id=100  |
                                                       + rrc  v
                                       {EncryptedExtensions}  ^  Server
                                        {CertificateRequest}  v  Params
                                               {Certificate}  ^
                                         {CertificateVerify}  | Auth
                                <--------         {Finished}  v

     ^ {Certificate}
Auth | {CertificateVerify}
     v {Finished}               -------->
       [Application Data]       <------->  [Application Data]

              +  Indicates noteworthy extensions sent in the
                 previously noted message.

              *  Indicates optional or situation-dependent
                 messages/extensions that are not always sent.

              {} Indicates messages protected using keys
                 derived from a [sender]_handshake_traffic_secret.

              [] Indicates messages protected using keys
                 derived from [sender]_application_traffic_secret_N.
```

Figure 1: Message Flow for Full TLS Handshake

Once a connection has been established the client and the server exchange application payloads protected by DTLS with an unilaterally used CIDs. In our case, the client is requested to use CID 100 for records sent to the server.

At some point in the communication interaction the IP address used by the client changes and, thanks to the CID usage, the security context to interpret the record is successfully located by the server. However, the server wants to test the reachability of the client at his new IP address.

```
Client                                                  Server
------                                                  ------

Application Data            ========>
<CID=100>
Src-IP=A
Dst-IP=Z
                            <========        Application Data
                                                 Src-IP=Z
                                                 Dst-IP=A


                        <<------------->>
                        <<   Some      >>
                        <<   Time      >>
                        <<   Later     >>
                        <<------------->>


Application Data            ========>
<CID=100>
Src-IP=B
Dst-IP=Z

                                       <<< Unverified IP
                                           Address B >>

                            <--------  Return Routability Check
                                       path_challenge(cookie)
                                              Src-IP=Z
                                              Dst-IP=B

Return Routability Check    -------->
path_response(cookie)
Src-IP=B
Dst-IP=Z

                                       <<< IP Address B
                                           Verified >>


                            <========        Application Data
                                                 Src-IP=Z
                                                 Dst-IP=B

            Figure 2: Return Routability Example
```

6. Security and Privacy Considerations

Note that the return routability checks do not protect against flooding of third-parties if the attacker is on-path, as the attacker can redirect the return routability checks to the real peer (even if those datagrams are cryptographically authenticated). On-path adversaries can, in general, pose a harm to connectivity.

7. IANA Considerations

IANA is requested to allocate an entry to the TLS ContentType registry, for the return_routability_check(TBD2) defined in this document. The return_routability_check content type is only applicable to DTLS 1.2 and 1.3.

IANA is requested to allocate the extension code point (TBD1) for the rrc extension to the TLS ExtensionType Values registry as described in Table 1.

Value	Extension Name	TLS 1.3	DTLS-Only	Recommended	Reference
TBD1	rrc	CH, SH	Y	N	RFC-THIS

Table 1: rrc entry in the TLS ExtensionType Values registry

8. Open Issues

Issues against this document are tracked at https://github.com/tlswg/dtls-rrc/issues

9. Acknowledgments

We would like to thank Achim Kraus, Hanno Becker, Hanno Boeck, Manuel Pegourie-Gonnard, Mohit Sahni and Rich Salz for their input to this document.

10. Normative References

[I-D.ietf-tls-dtls-connection-id] Rescorla, E., Tschofenig, H., Fossati, T., and A. Kraus, "Connection Identifiers for DTLS 1.2", Work in Progress, Internet-Draft, draft-ietf-tls-dtls-connection-id-13, 22 June 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-dtls-connection-id-13>.

[I-D.ietf-tls-dtls13] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-dtls13-43, 30 April 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-dtls13-43>.

[RFC2119]
Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[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/rfc/rfc8446>.

Appendix A. History

RFC EDITOR: PLEASE REMOVE THIS SECTION

draft-ietf-tls-dtls-rrc-02

* Undo the TLS flags extension for negotiating RRC, use a new extension type

draft-ietf-tls-dtls-rrc-01

* Use the TLS flags extension for negotiating RRC
* Enhanced IANA consideration section
* Expanded example section
* Revamp message layout:
  - Use 8-byte fixed size cookies
  - Explicitly separate path challenge from response

draft-ietf-tls-dtls-rrc-00

* Draft name changed after WG adoption

draft-tschofenig-tls-dtls-rrc-01

* Removed text that overlapped with draft-ietf-tls-dtls-connection-id

draft-tschofenig-tls-dtls-rrc-00

* Initial version

Authors' Addresses

Hannes Tschofenig (editor)
Arm Limited

Email: hannes.tschofenig@arm.com

Thomas Fossati
Arm Limited

Email: thomas.fossati@arm.com