TLS
Internet-Draft
Updates: 6347 (if approved)
Intended status: Standards Track
Expires: January 9, 2020

T. Fossati
H. Tschofenig, Ed.
Arm Limited

July 08, 2019

Return Routability Check for DTLS 1.2 and DTLS 1.3
draft-tschofenig-tls-dtls-rrc-00

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.

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 January 9, 2020.

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 content NAT rebinding is a common reason 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.

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.

An on-path adversary could intercept and modify the source IP address (and the source port). Even if receiver checks the authenticity and freshness of the packet, the recipient is fooled into changing the CID-to-IP/port association. This attack is possible because the network and transport layer identifiers, such as source IP address and source port numbers, are not integrity protected and authenticated by the DTLS record layer.

This attack makes strong assumptions on the attacker's abilities, and moreover it only misleads the peer until the next message gets through un-intercepted.

A return routability check (RRC) 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 a certain degree of confidence to the receiving peer that the sending peer is reachable at the indicated address and port.

Without such a return routability check, an adversary can redirect traffic towards a third party or a black hole.

While an equivalent check can be performed at the application layer (modulo the DTLS API exposing the address update event to the calling application), it is advantageous to offer this functionality at the DTLS layer. Section 3 describes the application layer procedure and Section 4 specifies a new message to perform this return routability check.

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 solutions defined for DTLS 1.2 [I-D.ietf-tls-dtls-connection-id] and for DTLS 1.3 [I-D.ietf-tls-dtls13].

3. Application Layer Return Routability Check

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). This is to ensure that a man-on-the-middle attacker that sends a datagram

with a different source address/port on an existing CID session does not successfully manage to re-route any return traffic.

Furthermore, when using CID, anti-replay protection MUST be enabled. This is to ensure that a man-on-the-middle attacker sending a previously captured record with a modified source IP address and port will not be able to successfully pass the above check (since the datagram is very likely discarded on receipt - if it falls outside the replay window).

The two countermeasures cannot complete stop a man-in-the-middle attacker who performs a DoS on the sender or uses the receiver as as backscatter source for a DDoS attack. For a more generic protection, a return routability check is needed.

It is RECOMMENDED that implementations of the CID functionaliy described in [I-D.ietf-tls-dtls-connection-id] and in [I-D.ietf-tls-dtls13] added peer address update events to their APIs. Applications can then use these events as triggers to perform an application layer return routability check, for example one that is based on successful exchange of minimal amount of ping-pong traffic with the peer.

4. The Return Routability Check Message

```
enum {
    invalid(0),
    change_cipher_spec(20),
    alert(21),
```

```
          handshake(22),
          application_data(23),
          heartbeat(24),  /* RFC 6520 */
          return_routability_check(TBD), /* NEW */
          (255)
      } ContentType;
```

The newly introduced return_routability_check message contains a cookie. The semantic of the cookie is similar to the cookie used in the HelloRetryRequest message defined in [RFC8446].

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

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

1. A cookie is placed in the return_routability_check message;

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 it back;

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

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

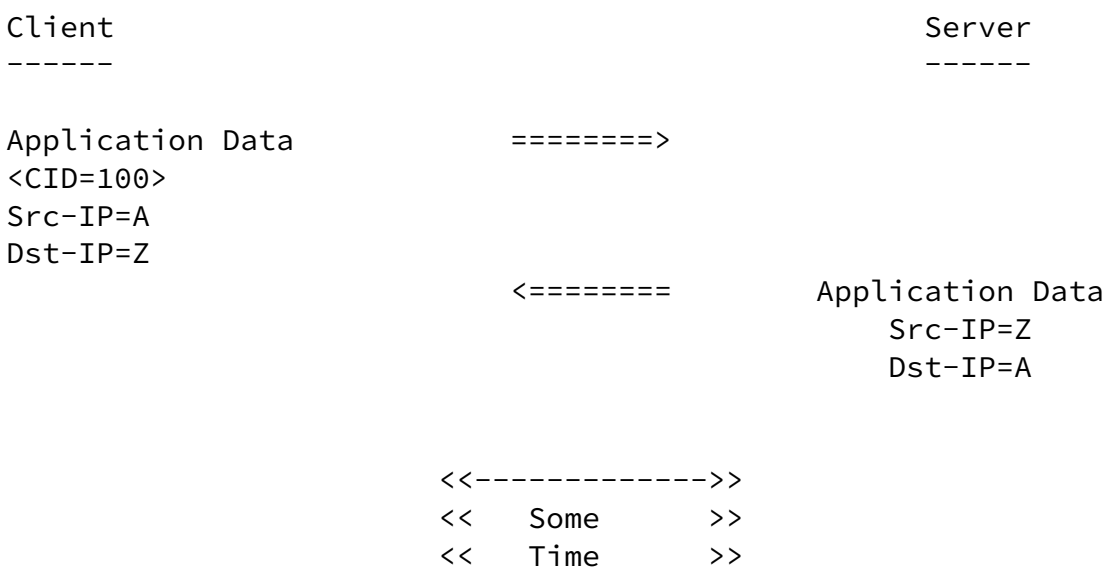
After this point, any pending send operation is resumed to the bound peer address.

```
struct {
    opaque cookie<1..2^16-1>;
} Cookie;

struct {
    Cookie cookie;
} return_routability_check;
```

5. RRC Example

The example shown in Figure 1 illustrates a client and a server exchanging application payloads protected by DTLS with an unilaterally used CIDs. 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, to avoid being abused (e.g., as an amplifier) by an attacker impersonating the client.



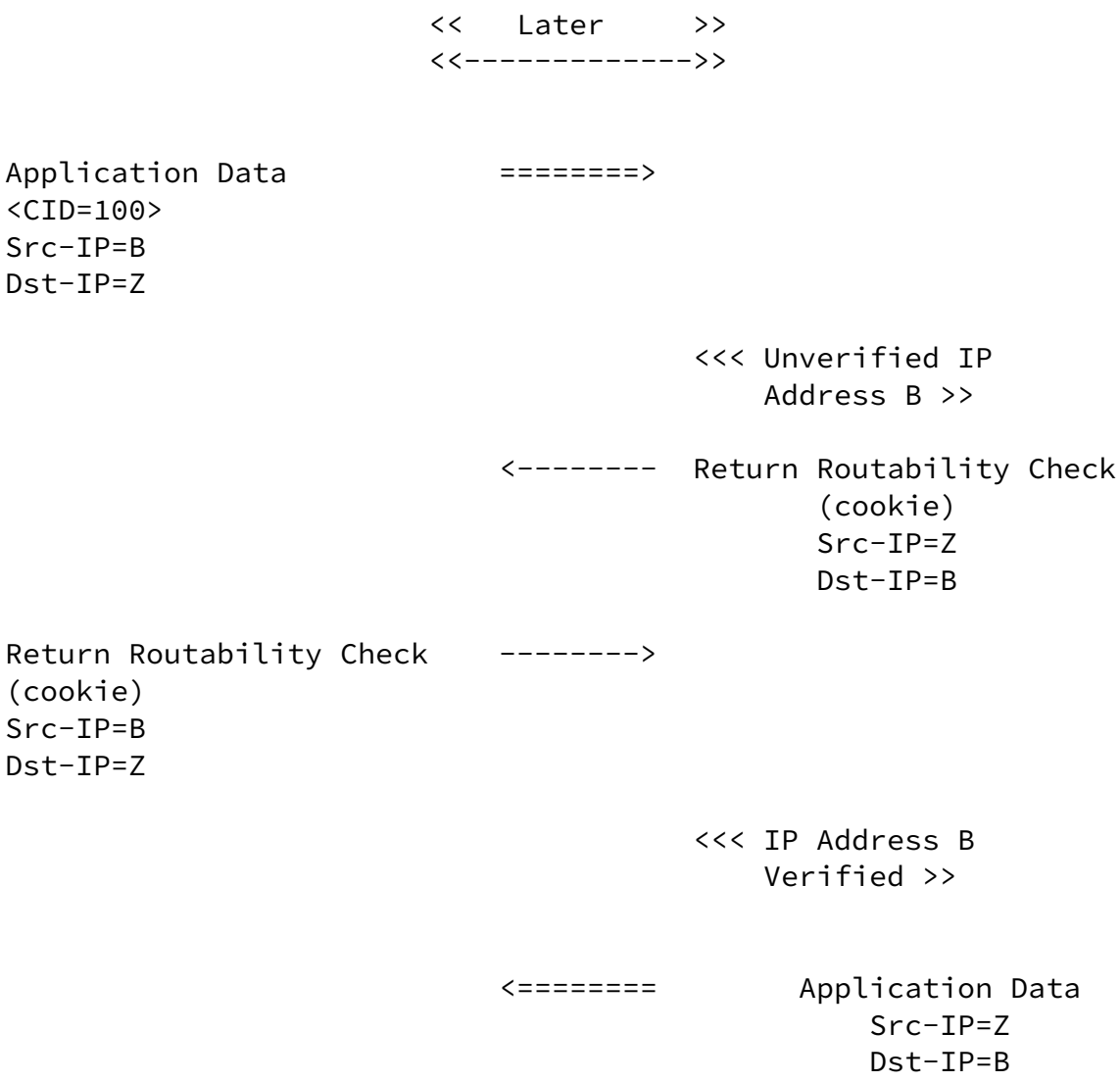


Figure 1: Return Routability Example

6. Security and Privacy Considerations

As all the datagrams in DTLS are authenticated, integrity and confidentiality protected there is no risk that an attacker undetectably modifies the contents of those packets. The IP addresses in the IP header and the port numbers of the transport layer are, however, not authenticated. With the introduction of the CID, care must be taken to test reachability of a peer at a given IP

address and port.

Note that the return routability checks do not protect against third-party flooding if the attacker is along the path, as the attacker can forward the return routability checks to the real peer (even if those datagrams are cryptographically authenticated).

7. IANA Considerations

IANA is requested to allocate an entry to the existing TLS "ContentType" registry, for the return_routability_check(TBD) defined in this document.

8. Open Issues

- Should the return routability check use separate sequence numbers and replay windows?

- Should the heartbeat message be re-used instead of the proposed new message exchange?

9. References

9.1. Normative References

[I-D.ietf-tls-dtls-connection-id]
Rescorla, E., Tschofenig, H., and T. Fossati, "Connection Identifiers for DTLS 1.2", draft-ietf-tls-dtls-connection-id-05 (work in progress), May 2019.

[I-D.ietf-tls-dtls13]
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", draft-ietf-tls-dtls13-31 (work in progress), March 2019.

[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/info/rfc2119>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security

(TLS) Protocol Version 1.2", RFC 5246,
DOI 10.17487/RFC5246, August 2008,
<https://www.rfc-editor.org/info/rfc5246>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[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/info/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/info/rfc8446>.

9.2. URIs

[1] mailto:tls@ietf.org

[2] https://www1.ietf.org/mailman/listinfo/tls

[3] https://www.ietf.org/mail-archive/web/tls/current/index.html

Appendix A. History

RFC EDITOR: PLEASE REMOVE THE THIS SECTION

- Initial version

Appendix B. Working Group Information

RFC EDITOR: PLEASE REMOVE THE THIS SECTION

The discussion list for the IETF TLS working group is located at the e-mail address tls@ietf.org [1]. Information on the group and information on how to subscribe to the list is at https://www1.ietf.org/mailman/listinfo/tls [2]

Archives of the list can be found at: https://www.ietf.org/mail-archive/web/tls/current/index.html [3]

Appendix C. Acknowledgements

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

Authors' Addresses

Thomas Fossati
Arm Limited

EMail: thomas.fossati@arm.com

Hannes Tschofenig (editor)
Arm Limited

EMail: hannes.tschofenig@arm.com