TCP Maintenance and Minor Extensions
Internet-Draft
Intended status: Standards Track
Expires: 2 September 2025

M. Boucadair
Orange
T. Reddy
Nokia
J. Xing
Tencent

1 March 2025

TCP RST Diagnostic Payload
draft-boucadair-tcpm-rst-diagnostic-payload-11

Abstract

This document specifies a diagnostic payload format returned in TCP RST segments. Such payloads are used to share with an endpoint the reasons for which a TCP connection has been reset. Sharing this information is meant to ease diagnostic and troubleshooting.

Discussion Venues

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

Discussion of this document takes place on the TCP Maintenance and Minor Extensions mailing list (tcpm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tcpm/.

Source for this draft and an issue tracker can be found at https://github.com/boucadair/draft-boucadair-tcpm-rst-diagnostic-payload.

Status of This Memo

Table of Contents

1. Introduction

A TCP connection [RFC9293] can be reset by a peer for various reasons, e.g., received data does not correspond to an active connection. Also, a TCP connection can be reset by an on-path service function (e.g., Carrier Grade NAT (CGN) [RFC6888], NAT64 [RFC6146], or firewall) for several reasons. Typically, a Network Address Translator (NAT) function can generate an RST segment to notify an endpoint upon the expiry of the lifetime of the corresponding mapping entry or because an RST segment was received from a peer (Section 2.2 of [RFC7857]).

A TCP connection can also be closed by a user or an application at any time. However, the peer that receives an RST segment does not have any hint about the reason that led to terminating the connection. Likewise, the application that relies upon such a TCP connection may not easily identify the reason for the connection closure. Troubleshooting such events at the remote side of the connection that receives the RST segment may not be trivial.

This document fills this void by specifying a format of the diagnostic payload that is returned in an RST segment. Returning such data is consistent with the provision in Section 3.5.3 of [RFC9293] for RST segments, especially:

> "TCP implementations SHOULD allow a received RST segment to include data (SHLD-2)."

This document does not change the conditions under which an RST segment is generated (Section 3.5.2 of [RFC9293]).

The generic procedure for processing an RST segment is specified in Section 3.5.3 of [RFC9293]. Only the deviations from that procedure to insert and validate a diagnostic payload is provided in Section 3. Section 4 provides a set of examples to illustrate the use of TCP RST diagnostic payloads.

This document specifies the format and the overall approach to ease maintaining the list of codes while allowing for adding new codes as needed in the future and accommodating any existing vendor-specific codes. An initial version of error codes is available in Table 2. However, the authoritative source to retrieve the full list of error codes is the IANA-maintained registry (Section 5.2).

Preliminary investigation based on some major CGN vendors revealed that RSTs with data are not discarded and are translated according to any matching mapping entry.

2. Conventions and Definitions

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 makes use of the terms defined in Section 4 of [RFC9293].

3. RST Diagnostic Payload

The RST diagnostic payload MUST be encoded using Concise Binary Object Representation (CBOR) Sequence [RFC8742]. The Concise Data Definition Language (CDDL) [RFC8610] for the diagnostic payload is shown in Figure 1.

```
; This defines an array, the elements of which are to be used
; in a CBOR Sequence. There is exactly one occurrence.
diagnostic-payload = [magic-cookie, reason]
; Magic cookie to identify a payload that follows this specification
magic-cookie = 12345
; Reset reason details:
reason= {
  ? reason-code: uint,
  ? pen:uint,
  ? reason-description: tstr,
}
; Map Keys
reason-code = 1
pen = 2
reason-description = 3
```

Figure 1: Structure of the RST Diagnostic Payload

The RST diagnostic payload comprises a magic cookie that is used to unambiguously identify an RST payload that follows this specification. It MUST be set to the RFC number to be assigned to this document.

> Note to the RFC Editor: Please replace "12345" with the RFC number assigned to this document.

All parameters in the reason component of an RST diagnostic payload are mapped to their CBOR key values as specified in Section 5.1. The description of these parameters is as follows:

reason-code: This parameter takes a value from an available registry such as the "TCP Failure Causes" registry (Section 5.2).

pen: Includes a Private Enterprise Number [Private-Enterprise-Numbers]. This parameter MAY be included when the reason code is not taken from the IANA-maintained registry (Section 5.2), but from a vendor-specific registry.

reason-description: Includes a brief description of the reset reason

encoded as UTF-8 [RFC3629]. This parameter MUST NOT be included if a reason code is supplied. This parameter is useful only for reset reasons that are not yet registered or for application-specific reset reasons.

At least one of "reason-code" and "reason-description" parameters MUST be included in an RST diagnostic payload. The "pen" parameter MUST be omitted if a reason code from the IANA-maintained registry (Section 5.2) fits the reset case.

Malformed RST diagnostic payload messages that include the magic cookie MUST be silently ignored by the receiver.

A peer that receives a valid diagnostic payload may pass the reset reason information to the local application in addition to the information (MUST-12) described in Section 3.6 of [RFC9293]. That information may also be logged locally, unless a local policy specifies otherwise. How the information is passed to an application and how it is stored locally is implementation-specific.

Per Section 3.6 of [RFC9293], one or more RST segments can be sent to reset a connection. Whether a TCP endpoint elects to send more than one RST with only a subset of them that include the diagnostic payload is implementation-specific.

4. Some Examples

To ease readability, the CBOR diagnostic notation (Section 8 of [RFC8949]) with the parameter names rather than their CBOR key values in Section 5.1 is used in Figures 3, 4, 5, and 6.

Figure 2 depicts an example of an RST diagnostic payload that is generated to inform the peer that the TCP connection is reset because an ACK was received from that peer while the connection is still in the LISTEN state (Section 3.10.7.2 of [RFC9293]).

```
19 3039 # unsigned(12345)
A1    # map(1)
   01 # unsigned(1)
   02 # unsigned(2)
```

Figure 2: Example of an RST Diagnostic Payload with Reason Code (CBOR Encoding)

Figure 3 depicts the same RST diagnostic payload as the one shown in Figure 2 but following the CBOR diagnostic notation.

```
[
  12345,
  {
    1: 2
  }
]
```

Figure 3: Example of an RST Diagnostic Payload with Reason Code (Diagnostic Notation)

Figure 4 shows an example of an RST diagnostic payload that includes a free description to report a case that is not covered by an appropriate code from the IANA-maintained registry (Section 5.2).

```
[
  12345,
  {
    3: "brief human-readable description"
  }
]
```

Figure 4: Example of an RST Diagnostic Payload with Reason Description (Diagnostic Notation)

An RST diagnostic payload may also be sent by an on-path service function. For example, the following diagnostic payload is returned by a NAT function upon expiry of the mapping entry to which the TCP connection is bound (Figure 5).

```
[
  12345,
  {
    1: 8
  }
]
```

Figure 5: Example of an RST Diagnostic Payload to Report Connection Timeout (Diagnostic Notation)

Figure 6 illustrates an RST diagnostic payload that is returned by a peer that resets a TCP connection for a reason code 1234 defined by a vendor with the private enterprise number 32473.

```
[
  12345,
  {
    1: 1234,
    2: 32473
  }
]
```

Figure 6: Example of an RST Diagnostic Payload to Report Vendor-Specific Reason Code (Diagnostic Notation)

Figure 6 uses the Enterprise Number 32473 defined for documentation use [RFC5612].

5. IANA Considerations

5.1. RST Diagnostic Payload CBOR Key Values

IANA is requested to create a new registry titled "RST Diagnostic Payload CBOR Key Values" under the "Transmission Control Protocol (TCP) Parameters" registry group [IANA-TCP].

The key value MUST be an integer in the 1-255 range.

The assignment policy for this registry is "IETF Review" (Section 4.8 of [RFC8126]).

The structure of this subregistry and the initial values are provided in Table 1.

Parameter Name	CBOR Key	CBOR Major Type & Information	Reference
reason-code	1	0 unsigned	[ThisDocument]
pen	2	0 unsigned	[ThisDocument]
reason-description	3	3 text string	[ThisDocument]

Table 1: Initial CBOR Keys

5.2. New Registry for TCP Failure Causes

This document requests IANA to create a new registry entitled "TCP Failure Causes" under the "Transmission Control Protocol (TCP) Parameters" registry group [IANA-TCP].

Values are taken from the 1-65535 range.

The assignment policy for this registry is "Expert Review" (Section 4.5 of [RFC8126]).

The designated experts may approve registration once they checked that the new requested code is not covered by an existing code and if the provided reasoning to register the new code is acceptable. A registration request may supply a pointer to a specification where that code is defined. However, a registration may be accepted even if no permanent and readily available public specification is available.

The registry is initially populated with the values listed in Table 2.

Value	Description	Specification (if available)
1	Illegal Option	Section 3.1 of [RFC9293]
2	Desynchronized state	Section 3.5.1 of [RFC9293]
3	New data is received after CLOSE is called	Sections 3.6.1 and 3.10.7.1 of [RFC9293]
4	ABORT Process	Section 3.10.5 of [RFC9293]
5	Unexpected ACK received by non-synchronized state connection	Section 3.10.7 of [RFC9293]
6	Unexpected SYN in the window	Section 3.10.7 of [RFC9293]
7	Unexpected security compartment	Appendix A.1 of [RFC9293]
8	Malformed Message	[ThisDocument]
9	Not Authorized	[ThisDocument]
10	Resource Exceeded	[ThisDocument]
11	Network Failure	[ThisDocument]

12	Reset received from he peer	[ThisDocument]
13	Destination Unreachable	[ThisDocument]
14	Connection Timeout	[ThisDocument]
15	Too much outstanding data	Section 3.6 of [RFC8684]
16	Unacceptable performance	Section 3.6 of [RFC8684]
17	Middlebox interference	Section 3.6 of [RFC8684]

Table 2: Initial TCP Failure Causes

Note that codes in the 8-14 range can be used by service functions (Carrier Grade NAT (CGN), firewall, proxy, etc.).

6. Security Considerations

[RFC9293] discusses TCP-related security considerations. In particular, RST-specific attacks and their mitigations are discussed in Section 3.10.7.3 of [RFC9293].

In addition to these considerations, it is RECOMMENDED to control the size of acceptable diagnostic payload and keep it as brief as possible. The RECOMMENDED acceptable maximum size of the RST diagnostic payload is 255 octets.

Also, it is RECOMMENDED to avoid leaking privacy-related information as part of the diagnostic payload (e.g., including a description such as "user X resets explicitly the connection" is not recommended). The "reason-description" string, when present, MUST NOT include any private information that an observer would not otherwise have access to.

The presence of vendor-specific reason codes (Section 3) may be used to fingerprint hosts. Such a concern does not apply if the reason codes are taken from the IANA-maintained registry. Implementers are, thus, encouraged to register new codes within IANA instead of maintaining specific registries.

The reason description, when present, MUST NOT be displayed to end users but is intended to be consumed by applications. Such a description may carry a malicious message to mislead the end-user.

7. References

7.1. Normative References

[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>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/rfc/rfc3629>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[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>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/rfc/rfc8610>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <https://www.rfc-editor.org/rfc/rfc8684>.

[RFC8742] Bormann, C., "Concise Binary Object Representation (CBOR) Sequences", RFC 8742, DOI 10.17487/RFC8742, February 2020, <https://www.rfc-editor.org/rfc/rfc8742>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

[RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/rfc/rfc9293>.

7.2. Informative References

[IANA-TCP] "Transmission Control Protocol (TCP) Parameters", <https://www.iana.org/assignments/tcp-parameters/tcp-parameters.xhtml#>.

[Private-Enterprise-Numbers] "Private Enterprise Numbers", May 2020, <https://www.iana.org/assignments/enterprise-numbers>.

[RFC5612] Eronen, P. and D. Harrington, "Enterprise Number for Documentation Use", RFC 5612, DOI 10.17487/RFC5612, August 2009, <https://www.rfc-editor.org/rfc/rfc5612>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/rfc/rfc6146>.

[RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <https://www.rfc-editor.org/rfc/rfc6888>.

[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/rfc/rfc7252>.

[RFC7857] Penno, R., Perreault, S., Boucadair, M., Ed., Sivakumar, S., and K. Naito, "Updates to Network Address Translation (NAT) Behavioral Requirements", BCP 127, RFC 7857, DOI 10.17487/RFC7857, April 2016, <https://www.rfc-editor.org/rfc/rfc7857>.

Acknowledgments

The "diagnostic payload" name is inspired by Section 5.5.2 of [RFC7252] that was cited by Carsten Bormann in the tcpm mailing list.

Thanks to Jon Shallow for the comments. Thanks also to Li Jinghui for the discussion.

Authors' Addresses

Mohamed Boucadair
Orange
Email: mohamed.boucadair@orange.com

Tirumaleswar Reddy
Nokia
India
Email: kondtir@gmail.com

Jason Xing
Tencent
Email: kerneljasonxing@gmail.com