TCP RST Diagnostic Payload

draft-boucadair-tcpm-rst-diagnostic-payload

IETF#124, November 2025, Montreal

M. Boucadair , T. Reddy, & **J. Xing**

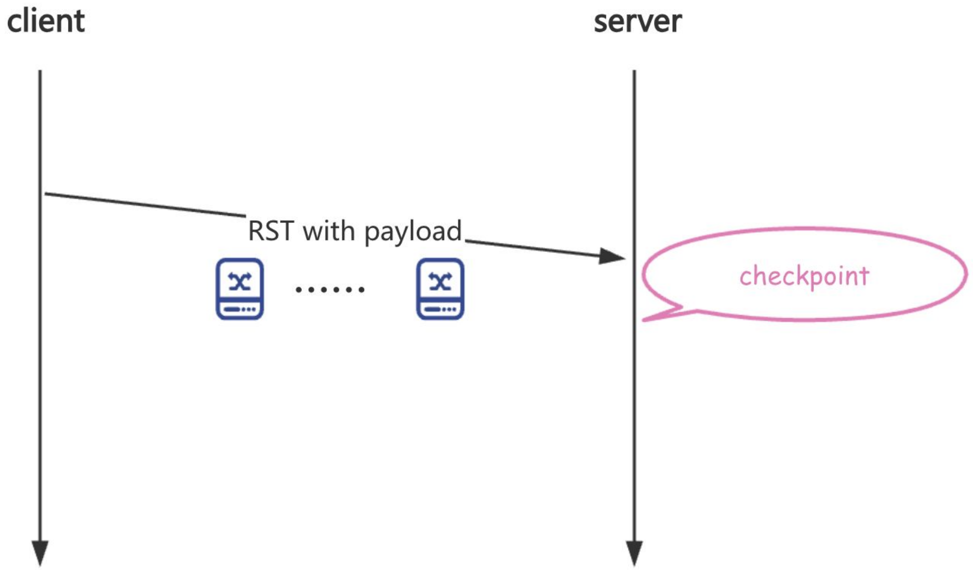
Pending Issues from IETF 122

- More experiments are needed to ensure sufficient data/validity
- Various middle box scenarios are needed to verify if any packet drop instance occurs

Current Experimental Status

- The implementation targets Linux 6.16.
- The whole experiments have been conducted and verified at Tencent among the real workload over six months.
  - softwares: internal DPDKs for gateway (L4 and L7)
  - drivers: virtio_net, i40e, ixgbe, mln5
  - kernels: 6.16 and a few other versions
  - distros: Ubuntu, CentOS
  - countries: China-to-Singapore (RTT > 30ms) and China-to-Germany (RTT > 200ms)

How to verify?



Expected behavior
- Client sends an RST packet with data
- Server receives and parses and handles the packet successfully w/o RST payload feature.

tcpdump Verification (1)

Commit 674fea6

yuguy committed on Jul 29, 2000

```
Changes from Kevin Steves to print, in an RST segment, the payload as
text, as per RFC 1122:

        4.2.2.12  RST Segment: RFC-793 Section 3.4

           A TCP SHOULD allow a received RST segment to include data.

           DISCUSSION
                It has been suggested that a RST segment could contain
                ASCII text that encoded and explained the cause of the
                RST.  No standard has yet been established for such
                data.

as some TCP implementations (e.g., HP-UX 11.0) do that.

Clean up the code that prints the payload of non-RST segments.
```

```
+ /*
+  * RFC1122 says the following on data in RST segments:
+  *
+  *         4.2.2.12  RST Segment: RFC-793 Section 3.4
+  *
+  *            A TCP SHOULD allow a received RST segment to include data.
+  *
+  *            DISCUSSION
+  *                 It has been suggested that a RST segment could contain
+  *                 ASCII text that encoded and explained the cause of the
+  *                 RST.  No standard has yet been established for such
+  *                 data.
+  *
+  */
```

- In 2000, **tcpdump already reserved the RST payload feature** as above which adheres to standard RFC 1122
- Corresponding link can be seen at https://github.com/the-tcpdump-group/tcpdump/commit/674fea664da96c

tcpdump Verification (2)

```
[kernelxing@VM-0-2-tencentos net-next]$ sudo tcpdump 'tcp[tcpflags] & tcp-rst != 0' and 'tcp[tcpflags] & tcp-ack != 0' -X -nn -vv -S
dropped privs to tcpdump
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:12:39.096715 IP (tos 0x0, ttl 64, id 26362, offset 0, flags [DF], proto TCP (6), length 1052)
    9.134.0.2.5201 > 9.134.121.177.59746: Flags [R.], cksum 0x538c (correct), seq 2695270080:2695271080, ack 3089774739, win 7990, op
        0x0000:  4500 041c 66fa 4000 4006 4323 0986 0002  E...f.@.@.C#....
        0x0010:  0986 79b1 1451 e962 a0a6 8ec0 b82a 3893  ..y..Q.b.....*8.
        0x0020:  8014 1f36 538c 0000 0101 080a 8da3 7902  ...6S.........y.
        0x0030:  4236 019b 0b00 0000 0000 0000 0000 0000  B6..............
        0x0040:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0050:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0060:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0070:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0080:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0090:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x00a0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x00b0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x00c0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x00d0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x00e0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x00f0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

- It has the following suffix string with the patch applied as expected:
  - '[RST+ ^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@...]'
- It accurately prints out the RST reason that the client sets in the payload consciously.
- Tcpdump does NOT intercept the RST packet with data.

Kernel Implementation

- The overall design in Linux is to pre-allocate a large enough zeroed buffer, put a reset reason code in the first byte and sent it out to verify whether the RST with payload can be possibly declined by any equipment in between two sides and the other side successfully parses the RST with payload.
  - Payload Attachment: Allocate a 1000-byte data payload attached to all generated RST packets.
  - Reason Code Encoding: The first byte of the payload is used to store a predefined reset reason code that is listed in include/net/rstreason.h file, while the remainder of the payload is zero-padded.
  - Handling of Reset Types: The implementation distinguishes between the two primary reset scenarios in tcp_send_active_reset() and tcp_v4_send_reset() respectively:
- Link: https://datatracker.ietf.org/doc/draft-boucadair-tcpm-rst-diagnostic-payload/#:~:text=RST%20with%20payload.-,A.1.%20%20Implementation,-The%20following%20implementation

Next Steps

- Request WG Adoption