

Zero Checksum for SCTP

draft-tuexen-tsvwg-sctp-zero-checksum-02

Michael Tüxen (tuexen@fh-muenster.de)

Victor Boivie (boivie@google.com)

Florent Castelli (orphis@google.com)

Randell Jesup (randell-ietf@jesup.org)

Motivation

- SCTP uses CRC32c to detect errors during transmission. Motivated by SCTP/IPv4 and SCTP/IPv6.
- The CRC32c provides no value when SCTP/DTLS (WebRTC) is used but uses CPU resources for computing.
- Allow, in a backwards compatible way, to use zero as an incorrect CRC32c reducing the CPU resources needed.

packetdrill Example

```
0.0 socket(..., SOCK_STREAM, IPPROTO_SCTP) = 3
+0.0 fcntl(3, F_SETFL, O_RDWR|O_NONBLOCK) = 0
+0.0 setsockopt(3, IPPROTO_SCTP, SCTP_ACCEPT_ZERO_CHECKSUM, [1], 4) = 0
+0.0 connect(3, ..., ...) = -1 EINPROGRESS (Operation now in progress)
+0.0 > sctp: INIT[flgs=0, tag=1, a_rwnd=..., os=..., is=..., ts=1,
                ZERO_CHECKSUM_ACCEPTABLE[],
                SUPPORTED_ADDRESS_TYPES[types=[IPv4]]]
+0.0 < sctp(zero_checksum): INIT_ACK[flgs=0, tag=1, a_rwnd=1500, os=1, is=1, ts=1,
                STATE_COOKIE[len=4, val=...],
                ZERO_CHECKSUM_ACCEPTABLE[]]
+0.0 > sctp: COOKIE_ECHO[flgs=0, len=4, val=...]
+0.0 < sctp(zero_checksum): COOKIE_ACK[flgs=0]
+0.0 close(3)= 0
+0.0 > sctp(zero_checksum): SHUTDOWN[flgs=0, cum_ts=0]
+0.0 < sctp(zero_checksum): SHUTDOWN_ACK[flgs=0]
+0.0 > sctp(zero_checksum): SHUTDOWN_COMPLETE[flgs=0]
```

Changes

- Address comments from Claudio, Gorrry, Magnus, and Mike.
- Require correct CRC32c in addition to packets containing INIT chunks also for packets containing
 - COOKIE ECHO chunks
 - ASCONF chunks
- Allow stacks to accept OOTB packets with incorrect zero checksum.

Implementation Status

- Implemented in the FreeBSD kernel and the usrsctp userland stack.
- Currently being tested for integration into dcSCTP.
- Test suite for packetdrill available at <https://github.com/sctplab/zero-checksum>

Next Steps

- Address
 - anything required to be done by the authors before considered for WG adoption.
 - any additional feedback.