GDP - Bug #53

Segfault in gdp-reader.c; cleanup of signature buffer

09/23/2016 01:10 AM - Nitesh Mor

Status:	Closed	**Start date:**	09/23/2016
Priority:	Normal	**Due date:**	
Assignee:	Eric Allman	**% Done:**	0%
Category:	libgdp	**Estimated time:**	0.00 hour
Target version:			

Description

In the case of a log with signatures, an EOF causes a segmentation fault. Here's an example of how to reproduce it:

```
$ apps/gdp-reader -t  -f -1 edu.berkeley.eecs.swarmlab.device.c098e570008e
```

It is related to the re-use of the internal buffers. Running with a -D gdp.proto=50, one can see something like this right before the program crashes:

```
>>> _gdp_invoke(req=0x1695b70 rid=0): CMD_READ (70), gcl@0x1695df0
    datum @ 0x1695ce0: recno 9220238, len 0, no timestamp
_gdp_invoke: sending 70, retries=2
_gdp_invoke: waiting on 0x1695b70
_gdp_req_dispatch >>> NAK_C_NOTFOUND (196) [gcl->refcnt 2], req@0x1695b70:
    nextrec=0, numrecs=0, chan=0x1692d50
    postproc=(nil), sub_cb=(nil), udata=(nil)
    state=WAITING, stat=OK
    act_ts=2016-09-23T07:51:55.296667000Z
    flags=0x100<ON_CHAN_LIST>
    GCL@0x1695df0: XYD0I07GnCPDStIL74qRSjysTPjx4CpjjVMVjPAAeCA
    iomode = 1, refcnt = 2, reqs = (nil), nrecs = 9220237
    flags = 0xa<INCACHE,INUSE>
    PDU@0x1695c60:
    v=3, ttl=15, rsvd1=0, cmd=70=CMD_READ
    dst=XYD0I07GnCPDStIL74qRSjysTPjx4CpjjVMVjPAAeCA
    src=yTv93WOHsQOIjYNGuA7yFWeDw2eWoPQDAvWFBwaCxzA
    rid=0, olen=0, chan=(nil), seqno=0
    flags=0
    datum=0x1695ce0, recno=9220238, dbuf=0x1695d50, dlen=0
        ts=(none)
    sigmdalg=0x3, siglen=78, sig=0x7fa29c000ea0
    total header=88
    rPDU@0x7fa29c000db0:
    v=3, ttl=15, rsvd1=0, cmd=196=NAK_C_NOTFOUND
    dst=yTv93WOHsQOIjYNGuA7yFWeDw2eWoPQDAvWFBwaCxzA
    src=XYD0I07GnCPDStIL74qRSjysTPjx4CpjjVMVjPAAeCA
    rid=0, olen=8, chan=0x1692d50, seqno=0
    flags=0x2<HAS_RECNO>
    datum=0x7fa29c000e30, recno=9220238, dbuf=0x16970e0, dlen=0
        ts=(none)
    sigmdalg=0x3, siglen=0, sig=(nil)
    total header=96
nak_client: received NAK_C_NOTFOUND for CMD_READ
nak_client: reusing old datum for req 0x1695b70
   datum @ 0x1695ce0: recno 9220238, len 0, no timestamp
Segmentation fault
```

The field to notice in this debug-log is that in PDU, the field sig ought to be null (outgoing read request). A side question is: are there

any implications on what gets sent out on the network, if there is a non-null signature for a request that doesn't need signatures?

The trouble part that causes the segmentation fault is in source:gdp/gdp_proto.c#L277 in the function acknak, which incorrectly assumes that a PDU with signature will have an rPDU with signature. Also, there is no explicit else for the if in line 271, but I believe the memcpy in line 283 takes care of the situation where a PDU has null sig, but rPDU has non-null sig.

History

#1 - 09/23/2016 02:22 AM - Eric Allman

- Status changed from New to Feedback

The analysis is correct; the signature buffer reuse was broken. Fixed in commit:354ee62a.

We should really eliminate the datum->siglen field, as it is redundant with length(datum->sig). That's probably too radical a change to do right now.

#2 - 09/23/2016 10:06 AM - Nitesh Mor

- Status changed from Feedback to Closed

Seems to work. Closing.