OsmoBSC - Bug #3806

OsmoBSC accepts BSSAP with wrong length field

02/18/2019 01:17 PM - laforge

Status:	Stalled	**Start date:**	02/18/2019
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	40%
Category:	A interface		
Target version:			
Spec Reference:			

Description

As seen in #3805, OsmoBSC would happily accept BSSMAP CLEAR COMMAND messages with IEs that extend beyond the length field of the BSSAP header.

This is definitely wrong. We should

- parse the length field
- ensure we have a minimum of that number of bytes of payload as specified by the length field
- truncate the msgb to a payload length as specified

This way any additional garbage at the end of a message would simply be ignored, with us only parsing the specified "length" number of bytes.

Let's also make sure to add TTCN-3 tests for this, intentionally sending length field values too large and too short.

Once implemented in OsmoBSC, we should also implement it on the MSC side.

Related issues:

Related to OsmoMSC - Bug #3805: OsmoMSC sends invalid BSSMAP length field on ...	**Resolved**	**02/18/2019**

Associated revisions

Revision 023fc49e - 03/27/2019 08:01 AM - dexter

osmo_bsc_bssap: check bssmap length field

At the moment the length field of the bssmap header is not parsed.
Instead the length is computed out of the known header length and the
number of bytes received. This is prone to error, lets make sure that
extranous data at the end of a message is ignored by parsing the bssmap
length correctly.

Change-Id: Idef2e783d2377a2ad1f697ea4d26491a32b3e549
Related: OS#3806

History

#1 - 02/18/2019 01:18 PM - laforge

- Related to Bug #3805: OsmoMSC sends invalid BSSMAP length field on CSFB CLEAR COMMAND added

#2 - 03/18/2019 05:18 PM - dexter

- Status changed from New to In Progress

- % Done changed from 0 to 40

I have now integrated checking+truncating of the bssmap message length, there is no TTCN3 test yet.

https://gerrit.osmocom.org/#/c/osmo-bsc/+/13306 osmo_bsc_bssap: check bssamp length field
https://gerrit.osmocom.org/#/c/osmo-msc/+/13307 a_iface_bssap: check bssamp length field

#3 - 03/26/2019 08:40 AM - dexter

The two patches for the length check, which I proposed look good in review. However, I have problems creating with TTCN3 for this since TTCN3

seems to let me only generate valid CLEAR COMMANDs. The best would be if I could somehow send a hand crafted SCCP payload, however, I do not know if this is possible.

#4 - 03/26/2019 10:11 AM - laforge

- File bssap_adapter.diff added

On Tue, Mar 26, 2019 at 08:40:40AM +0000, dexter [REDMINE] wrote:

> The two patches for the length check, which I proposed look good in review. However, I have problems creating with TTCN3 for this since TTCN3 seems to let me only generate valid CLEAR COMMANDs. The best would be if I could somehow send a hand crafted SCCP payload, however, I do not know if this is possible.

1) Normal option

Normally, You'd need to use the SCCP_Emulation directly, without the BSSMAP_Emulation on top.

2) Hackish option

Alternatively, you could extend the BSSAP_Conn_PT (port betwene BSSMAP_Emulation and ConnHdlr)
to accept something like 'octetstring' (in addition to PDU_BSSAP, etc.) and extend the
BSSMAP_Emulation.main() function with a

```
    [] CLIENT.receive(octetstring:?) -> value oct {
        BSSAP.send(oct);
    }
```

you'd also have to extend BSSAP_CODEC_PT similarly, implementing somethin like a
"type record BSSAP_N_DATA_RAW_req" which uses "octetstring" instead of PDU_BSSAP.

However, I guess at that point it all becomes too complex. So I'd suggest to simply
go for yet another way:

3) BSSAP_Adapter without "ops"

When you call f_bssmap_init with an "omit" argument as BssmapOps, then you basically
get the entire SIGTRAN stack up to SCCP, but without any BSSAP/BSSMAP on top.

You would then have something like a

```
modulepar {
    BSSAP_Configuration mp_bssap_cfg := { ... }; /* like BSC_Tests.ttcn */
}

component RAW_SCCP_CT {
    BSSAP_Adapter g_ba;
    port SCCPasp_PT SCCP;
}

function f_init() runs on RAW_SCCP_CT {
    f_bssap_init(g_ba, mp_cfg, "RAW_SCCP", omit);
    connect(self:SCCP, g_ba.vc_SCCP:SCCP_SP_PORT);
}
```

from that point onwards, you can then use things like

```
testcase TC_foo() runs on RAW_SCCP_CT {
    f_init();
    SCCP.send(t_ASP_N_CONNECT_req(called, calling, omit, omit, '01020304'O, conn_id, omit));
    SCCP.receive(tr_ASP_N_CONNECT_res ....
```

You may need the attached patch to fix a bug in BSSAP_Adapter.ttcn

#5 - 09/04/2019 09:26 AM - laforge

- Assignee deleted (dexter)

#6 - 08/28/2020 08:15 PM - fixeria

- Status changed from In Progress to Stalled

Files