OsmoMGW - Bug #5119

mgcp_client.c should not assert on unexpected codec name in the input data

04/18/2021 03:42 PM - neels

Status: In Progress
Priority: High
Assignee: dexter
Category: 
Target version: 
Spec Reference: 

Description

see
https://git.osmocom.org/osmo-mgw/tree/src/libosmo-mgcp-client/mgcp_client.c?id=9ffaba7c1b0e3e44469e8d4c9e30dfe0e31f07f2#n1121

```
static int add_lco(struct msgb *msg, struct mgcp_msg *mgcp_msg)
{
    unsigned int i;
    int rc = 0;
    const char *codec;
    unsigned int pt;

    rc += msgb_printf(msg, "L:");
    if (mgcp_msg->ptime)
        rc += msgb_printf(msg, " p:%u", mgcp_msg->ptime);

    if (mgcp_msg->codecs_len) {
        rc += msgb_printf(msg, " a:");
        for (i = 0; i < mgcp_msg->codecs_len; i++) {
            pt = mgcp_msg->codecs[i];
            codec = get_value_string_or_null(osmo_mgcpc_codec_names, pt);
            OSMO_ASSERT(codec);
            rc += msgb_printf(msg, "%s", extract_codec_name(codec));
            if (i < mgcp_msg->codecs_len - 1)
                rc += msgb_printf(msg, ";");
        }
        rc += msgb_printf(msg, ",");
    }

    rc += msgb_printf(msg, " nt:IN\r\n");
    return rc;
}
```

Results in

```
20210418153207176 DCHAN DEBUG lchan_rtp(0-0-1-TCH_F-0)[0x6120000105a0]{WAIT_MGW_ENDPOINT_AVAILABLE}: state_chg to WAIT_MGW_ENDPOINT_AVAILABLE (lchan_rtp_fsm.c:119)
20210418153207176 DRLL DEBUG MGCP_CONN(msc0-conn1_subscr-IMSI-001019876543210)[0x6120000102a0]{ST_CRCX}: is child of mgw-endp(msc0-conn1_subscr-IMSI-001019876543210)[0x612000010420] (fsm.c:491)
20210418153207176 DRLL DEBUG MGCP_CONN(msc0-conn1_subscr-IMSI-001019876543210)[0x6120000102a0]{ST_CRCX}: Received Event EV_CRCX (mgcp_client_fsm.c:636)
20210418153207176 DRLL DEBUG MGCP_CONN(msc0-conn1_subscr-IMSI-001019876543210)[0x6120000102a0]{ST_CRCX}: MGW/CRCX: creating connection on MGW endpoint:rtpbridge/*@mgw... (mgcp_client_fsm.c:221)
Assert failed codec ..../..
```

06/02/2021 1/2
 Apparently an incorrect codec string in the CRCX input data leads to an OsmoBSC crash. OSMO_ASSERT() is the wrong error handling for input data.

The context of this is that I am working on the Channel Mode Modify code path in OsmoBSC. It is probably a bug in that code path that passes a wrong codec name, but still this should not crash the entire BSC.

Related issues:
Related to OsmoMGW - Bug #5123: coredump nightly mgw on 3g voicecall startup  Resolved  04/20/2021

History
#1 - 04/20/2021 01:33 PM - laforge
- Related to Bug #5123: coredump nightly mgw on 3g voicecall startup added

#2 - 04/20/2021 01:35 PM - laforge
- Priority changed from Normal to High

In general, no matter what happens at a remote implementation that sends packets to us, we must never OSMO_ASSERT(). This is a serious problem. OSMO_ASSERT() is to guard against conditions entirely under control of our implementation (mgw in this case).

Any remote user, even a malicious one, must always be ble to send us anything without us running into OSMO_ASSERT(). If a remote user can trigger this, it's a denial of service vulnerability.

#3 - 05/10/2021 10:15 AM - dexter
- Status changed from New to In Progress
- % Done changed from 0 to 90

I have replaced the ASSERT with a normal check that just returns an error code. The reason for the ASSERT might have been that the API user is expected to use only the constants from mgcp_client.h. However, it may still be that the input data is derived from some other source and not filled in by the API user manually. So errors in the input data are possible.

https://gerrit.osmocom.org/c/osmo-mgw/+/24182  mgcp_client: do not crash when lco/sdp cannot be generated