OsmoNITB - Bug #2429

SMSC: deiverSM message with bad user_message_reference

08/10/2017 10:37 AM - pespin

Status:	Closed	**Start date:**	08/10/2017
Priority:	Normal	**Due date:**	
Assignee:	pablo	**% Done:**	100%
Category:			
Target version:			
Spec Reference:			

Description

After recent changes (most probably between 18ca1ce2ea9f0c353e6886a9d05c61238bba9dc6 and ac8ca4cfd19a23131959e88be49b6c56738a38c0 both included), osmo-gsm-tester test "smpp/esme_ms_sms_storeforward" [1] fails due to unexpected user_message_reference being received.

In that test, the ESME sends a "submit_sm" with user_message_reference=2 to a yet-not-registered subscribed MS, and when that MS powers on and registers to the network, a "deliver_sm" is sent to the ESME but with user_message_reference=0.

I attach a packet trace I took on the main unit (manually on the "lo" iface due to current issue #2413), which shows the packets and confirm the description above.

[1] https://git.osmocom.org/osmo-gsm-tester/tree/suites/smpp/esme_ms_sms_storeforward.py

Related issues:

Related to OsmoNITB - Bug #2414: SMSC: deliverSM message with no user_message...	**New**	**07/31/2017**
Related to OsmoGSMTester - Bug #2413: osmo-gsm-tester: smpp: ESME traffic not...	**Closed**	**07/31/2017**
Related to OsmoNITB - Bug #2353: SMSC: ESME sending an sms with registered_de...	**Closed**	**07/06/2017**
Related to OsmoGSMTester - Bug #2452: osmo-gsm-tester: ESME should send submi...	**New**	**08/18/2017**

History

#1 - 08/10/2017 10:38 AM - pespin

- Related to Bug #2414: SMSC: deliverSM message with no user_message_reference added

#2 - 08/10/2017 10:38 AM - pespin

- Related to Bug #2413: osmo-gsm-tester: smpp: ESME traffic not logged in nitb pcap file added

#3 - 08/10/2017 10:39 AM - pespin

- Related to Bug #2353: SMSC: ESME sending an sms with registered_delivery=0x01 not receiving delivery_sn message added

#4 - 08/10/2017 10:50 AM - pespin

- File osmo-nitb-log.txt added

I attach a logfile of osmo-nitb while running that test.

#5 - 08/10/2017 11:17 AM - pespin

- File osmo-nitb.pcap added

I attach pcap trace of osmo-nitb network part + GMSTAP in case it's useful.

#6 - 08/11/2017 10:44 AM - pablo

I can see a Delivery Receipt is being sent in osmo-nitb.txt

[[0;m[[1;38m20170810122201180 DLSMS <0024> gsm_04_11.c:664 id:00000001 sub:000 dlvrd:000 submit date:YYMMDDhhmm done date:YYMMDDhhmm stat:DELIVRD err:000 text:íò|^^><97>An¹^K$c<81>æm8^\4/»

So your tests are sending a SMS with sms->status_rep_req field set to 1 (I know this is python code, just telling how we express this in C), however,

we didn't support this before my patchset.

Two choices here:

1) You update tests to support "Delivery Receipts" handling as explained by 98849f7ddcbe8b094af5ee5efc30bd841ac12e04. I suspect your ESME code is handling a DELIVER_SM esm_class = Delivery Receipt as a real SMS, not a report. So osmo-gsm-tester, which I was not familiar with BTW, can now tests the Delivery Receipt path too. If you follow up longer path, then you would need to update this test code to support this too: a6eaeb88f5c81ce1b3802727a3fa087da0bc4e3f

2) You update your tests to send a sms->status_rep_req set to 0. This would be easier I guess to calm down this error.

#7 - 08/13/2017 05:04 PM - pespin

Yes I know we didn't support this before your patches, I actually discovered it a few weeks ago while adding SMPP tests to osmo-gsm-tester. See commit [1] for reference, in which I leave all the mechanism prepared to store "user_message_reference" ids we send and look for them when we receive a Deliver_SM message, then in the test be able to wait until we receive the Delivery receipt and do the validation. However, as the receipts were still not implemented at that time, I created issue #2353 (I think we can close it soon) and faked the wait function to return True for now:

```
    def receipt_was_received(self, umref):
        # return umref not in self.references_pending_receipt
        self.log('FIXME: wait_receipt disabled because receipts are not received, see OsmoNITB #2353')
        return True
```

So, as you can see, that path (delivery receipts) is already being checked in osmo-gsm-tester (check that the user_message_reference received is correct), but to have a PASS in the test for now as we know it's not implemented, I just return True when asking to wait for the receipt (deliver_sm) because it was never arriving before your patches and thus the "references_pending_receipt" list was never emptied.

However, after your patches, as you implement the feature (thanks for that btw), the returned user_message_reference in the deliver_sm is not correct, and then test validation fails as it finds out that user_message_reference received is not in the list of pending references.

According to "Optional Parameters for deliver_sm" in SMPP v3.4 documentation [2]:

```
user_message_reference:
A reference assigned by the  originating SME to the  message. In the case that the  deliver_sm is carrying an
 SMSC delivery receipt, an SME delivery acknowledgement or an SME  user acknowledgement (as indicated in the
esm_class field), the  user_message_reference parameter is set to the  message reference of the original messa
ge
```

And what the failing test is doing, is really similar to the simple case described in SMPP v3.4 documentation [2], Chapter "2.10.1 Store and Forward Message Mode", Figure 2-7: "Typical SMPP sequence for a registered store and forward message".

In summary: ESME sends a SUBMIT_SM message to an MS through SMSC with "Delivery receipt" enabled and user_message_reference=X, and then it expects to receive at some point a DELIVER_SM containing user_message_reference=X, and not user_message_reference=0 as it happens now.

So please, tell me if I am missing some important point or you can acknowledge that there's still an issue in the current implementation inside Openbsc after your patches.

[1] https://git.osmocom.org/osmo-gsm-tester/commit/?id=db0d8ab4fcfac67e76729241d92ca128d4526240
[2] http://opensmpp.org/specs/SMPP_v3_4_Issue1_2.pdf

#8 - 08/14/2017 11:59 AM - pablo

> ESME sends a SUBMIT_SM message to an MS through SMSC with "Delivery receipt" enabled and user_message_reference=X, and then it expects to receive at some point a DELIVER_SM containing user_message_reference=X, and not user_message_reference=0 as it happens now.

ESME should send a SUBMIT_SM with registered_delivery field set to 1, not a esm_class = "Delivery Receipt".

Then, once the SMS is delivered to the mobile station, the SMSC sends a DELIVER_SM with esm_class = "Delivery Receipt" to the ESME. This is not a real SMS, but a report.

The ESME needs to handle this report accordingly, basically it reacts to this "Delivery Receipt" in this way: It sends a SUBMIT_SM with esm_class = Delivery Acknowledgement to the SMSC.

Then, the SMSC sends a GSM03.40 Status-Report to the mobile station.

This is working here in my testbed, without message_reference this would not work at all since the mobile phone would be able to correlate what message is being acked. So I suspect there is a problem in your test infrastructure...

The commit you refer is doing **a lot of things** in one go... what exactly do you want me to see? The fact that you explain many things in your commit message shows that you should have split this in one patch per thing you describe. Just a proposal if you want to help people understand better what you're doing on those tests.

#9 - 08/14/2017 12:14 PM - pablo

- File esme-delivery-receipts.pcapng added

I'm attaching a pcap trace that you can inspect from my testbed.

#10 - 08/16/2017 08:22 AM - pespin

- File smpp-simple.pcap added

Hi Pablo, thanks for the pcap file. I just realized I actually attached the wrong pcap file which does not contain the SMPP traces. With openbsc e6222ef1acc9947dba82898a2371f48e4f3848e1 and latest libsmpp34, I'm still seeing the issue.

I simplified the test to the following:

1. Start openbsc, osmo-bts
2. Connect esme to openbsc
3. register the MS with openbsc (I see an Alert_notification message in SMPP trace)
4. I send an sms from esme to MS (Submit_sm, user_message_reference=0x1, "Delivery receipt request"=0x1 and with mode=DATAGRAM as I saw in your pcap trace, because I was using mode=Store&Forward before in the test).
5. I receive the Submit_sm resp from openbsc
6. I receive the Deliver_sm with mode=Default and type="Delivery Receipt", but with wrong user_message_reference.

   The ESME needs to handle this report accordingly, basically it reacts to this "Delivery Receipt" in this way: It sends a SUBMIT_SM with esm_class = Delivery Acknowledgement to the SMSC.

I see that I am missing this part, thanks for pointing it out, I'll implement that. However, the issue with receiving wrong user_message_reference appears before having to send this packets, which means this should not be related to the current issue.

I attach now the correct pcap trace containing only the smpp connection. I compared it with yours and I could not see anything indicating me a big difference in protocol side which could explain the issue. The only big difference is that in your trace there's a first deliver_sm packet sending a "Hello" message with the same user_message_reference.

#11 - 08/17/2017 06:42 AM - laforge

- Assignee set to pablo

#12 - 08/17/2017 08:46 AM - pablo

Message reference in your pcap file is 0x0001, which is weird. This should be 0x0100 if your intention is to use reference number 1, look:

```
print hex(socket.htons(0x0001))
```

0x100

SMPP specs say message reference is 16 bits field. However, message reference field in GSM03.40 spec is only 8 bytes.

So after ntohs() in OpenBSC, you should get back the reference number 1.

My impression is that your ESME code is buggy, not calling htons().

#13 - 08/17/2017 08:57 AM - pespin

- Assignee changed from pablo to pespin

Thanks for pointing that out, I didn't think about it, I'll check the code of python-smpplib to see if htons() is being used there.

I guess you meant 8 bits instead of 8 bytes. That also means I should take care of not using references bigger than 255 right?

#14 - 08/17/2017 09:58 AM - pablo

Oh right, 8 bits :-), so yes, that should be between 0 and 255.

#15 - 08/17/2017 02:38 PM - pespin

- Assignee changed from pespin to pablo

```
>>> n = 2
>>> print(struct.pack("H", n))  # native byte order (little endian)
b'\x02\x00'
>>> print(struct.pack("<H", n)) # little endian
b'\x02\x00'
>>> v = struct.pack(">H", n) # big endian
b'\x00\x02'
```

On your system print hex(socket.htons(0x0001)) = 0x1000 but it's still a little-endian machine, so it's \x00\x01 in memory.

Which means user_message_ref=1 is \x00\x01 in the wire, correct me if I'm wrong.

Quick look at the code of libsmpp34:
submit_sm.tlv:

```
if( inst_tlv-> tag == TLVID_user_message_reference ){\
    U16( inst_tlv->, value.val16, valueDec_16 );\
```

smpp34_unpack.c:77

```
#define U16( inst, par, _str ){\
    lenval = sizeof( uint16_t );\
    if( lenval > left ){\
        PUTLOG("[%s:%04X(%s)]", par, inst par,\
                                    "Value length exceed buffer length");\
        return( -1 );\
    }\
    memcpy(&inst par, aux, lenval);\
    left -= lenval; aux += lenval;\
    inst par = ntohs( inst par );\
    _str(inst par, dummy_b);\
    if( strcmp("", dummy_b) == 0 ){\
        PUTLOG("[%s:%04X(%s)]", par, inst par, "Invalid value");\
        return( -1 );\
    };\
    PUTLOG("[%s:%04X(%s)]", par, inst par, "OK");\
```

^ ntohs() is being called in there, which means after smpp34_unpack() is called, t->value.val16 is already in host byte order. I verified manually by adding printf printing the t->value.val16 value before and after ntohs() is called. Value seems correct beforehand, and it's wrong afterwards.

The the following patch is needed:

```
diff --git a/openbsc/src/libmsc/smpp_openbsc.c b/openbsc/src/libmsc/smpp_openbsc.c
index e656376b2..255211557 100644
--- a/openbsc/src/libmsc/smpp_openbsc.c
+++ b/openbsc/src/libmsc/smpp_openbsc.c
@@ -120,7 +120,7 @@ static int submit_to_sms(struct gsm_sms **psms, struct gsm_network *net,
                        }
                        break;
                case TLVID_user_message_reference:
-                       msg_ref = ntohs(t->value.val16);
+                       msg_ref = t->value.val16;
                        break;
                default:
                        break;
```

After that commit is applied, test in osmo-gsm-tester passes because value is handled correctly.

I also expect a similar issue during the opposite direction, but not that sure about that, I didn't verify it. See smpp34_pack():

```
smpp34_pack(uint32_t type, uint8_t *ptrBuf, int ptrSize, int *ptrLen, void* tt)
{
...
#define U16( inst, par, _str ) {\
```

```
    uint16_t v16 = htons(inst par);\
    lenval = sizeof(uint16_t);\
    if( lenval >= left ){\
        PUTLOG("[%s:%04X(%s)]", par, inst par,\
                                    "Value length exceed buffer length");\
        return( -1 );\
    };\
    _str(inst par,dummy_b);\
    if( strcmp("", dummy_b) == 0 ){\
        PUTLOG( "[%s:%04X(%s)]", par, inst par, "Invalid value");\
        return( -1 );\
    }\
    PUTLOG("[%s:%04X(%s)]", par, inst par, "OK");\
    memcpy(aux, &v16, lenval);\
    left -= lenval; aux += lenval;\
};
```

htons() is called there, which I guess means all values stored in TLVs before smpp34_pack() is called should be in native byte order. However, in smpp_openbsc.c I see:

```
void append_tlv_u16(tlv_t **req_tlv, uint16_t tag, uint16_t val)
{
    tlv_t tlv;

    memset(&tlv, 0, sizeof(tlv));
    tlv.tag = tag;
    tlv.length = 2;
    tlv.value.val16 = htons(val);
    build_tlv(req_tlv, &tlv);
}
```

Which is storing TLVs as network byte order. Probably wrong if after that smpp34_pack() is used.

My guess is that you didn't see these issues using smpp_mirror basically because it reuses a lot of same code which probably has same issues?

#16 - 08/17/2017 04:19 PM - pespin

the following change is needed:

```
@@ -436,7 +437,8 @@ void append_tlv_u16(tlv_t **req_tlv, uint16_t tag, uint16_t val)
        memset(&tlv, 0, sizeof(tlv));
        tlv.tag = tag;
        tlv.length = 2;
-       tlv.value.val16 = htons(val);
+       LOGP(DSMPP, LOGL_ERROR, "PESPIN append tag=%u val=%u\n", tag, val);
+       tlv.value.val16 = val;
        build_tlv(req_tlv, &tlv);
 }
```

It worked for me without it first because there's actually a bug when receiving integers in python-smpplib, I just sent a patch for that too:
https://github.com/podshumok/python-smpplib/pull/36/commits/5ad23c87a3932c209f2035fa04805f56c820ef4b

Not sure if smpp_mirror needs fixing too, I leave that up to you for now :-)

#17 - 08/18/2017 09:48 AM - pablo

> My guess is that you didn't see these issues using smpp_mirror basically because it reuses a lot of same code which probably has same issues?

Yes, I was reusing the existing code, so I was not aware that the libsmpp34 library already deals with endianess.

Would you submit patches for this to gerrit? I would ACK them.

Let me know, thanks for tracking down this issue.

#18 - 08/18/2017 10:16 AM - pespin

Sure, I'll prepare a patch probably today and add you as Reviewer.

It looks like smpp_mirror doesn't require any related change, there's no ntoh/hton being used there:

```
    if (t) {
        tlv_t tlv;

        memset(&tlv, 0, sizeof(tlv));
        tlv.tag = TLVID_user_message_reference;
        tlv.length = 2;
        tlv.value.val16 = t->value.val16;
        build_tlv(&submit.tlv, &tlv);
    }
```

which is basically the same code as in smpp_openbsc.c:append_tlv_u16() but without the faulty htons(). That makes smpp_mirror be correct.

#19 - 08/18/2017 10:51 AM - pespin

- Status changed from New to Feedback

Patch available in https://gerrit.osmocom.org/#/c/3554/

We can close the issue once it's merged and osmo-gsm-tester is green again.

I submitted too the other patch needed for python-smpplib. For now I patched manually the versions of python-smpplib running in both rnd and prod main units of osmo-gsm-tester. Once it is merged, we can install the new version of python-smpplib.

#20 - 08/18/2017 07:14 PM - pespin

- Related to Bug #2452: osmo-gsm-tester: ESME should send submit_sm as a response to deliver_sm added

#21 - 08/18/2017 07:17 PM - pespin

- Status changed from Feedback to Resolved

- % Done changed from 0 to 100

Patch was merged and jenkins is green again. I opened a new issue in #2452 to remember to implement the missing submit_sm response to the deliver_sm.

#22 - 10/11/2017 02:54 AM - laforge

- Status changed from Resolved to Closed

Files

bad_user_message_reference.pcap	2.27 KB	08/10/2017	pespin
osmo-nitb-log.txt	30 KB	08/10/2017	pespin
osmo-nitb.pcap	32.4 KB	08/10/2017	pespin
esme-delivery-receipts.pcapng	2.25 KB	08/14/2017	pablo
smpp-simple.pcap	2.32 KB	08/16/2017	pespin