Buffer Overflow in `ProcessRadioRxDone`

Moderate **mluis1** published **GHSA-7vv8-73pc-63c2** on Oct 5

Package

LoRaMac.c (LoRaMac-node)

Affected versions

< 4.7.0

Patched versions

4.7.0

Description

Buffer Overflow in `ProcessRadioRxDone`

Summary

Improper size validation of the incoming radio frames can lead to an 65280-byte out-of-bounds write.

Description

The function `ProcessRadioRxDone` implicitly expects incoming radio frames to have at least a `payload` of one byte or more.
An empty `payload` leads to a 1-byte out-of-bounds read of user controlled content when the payload buffer is reused. (This is for example the case in Zephyr.)

LoRaMac-node/src/mac/LoRaMac.c
Lines 876 to 881 in a166830

```
876     uint8_t *payload = RxDoneParams.Payload;
877     uint16_t size = RxDoneParams.Size;
878     int16_t rssi = RxDoneParams.Rssi;
879     int8_t snr = RxDoneParams.Snr;
880
881     uint8_t pktHeaderLen = 0;
```

LoRaMac-node/src/mac/LoRaMac.c
Line 941 in a166830

```
941      macHdr.Value = payload[pktHeaderLen++];
```

This allows an attacker to craft a `FRAME_TYPE_PROPRIETARY` frame with size `-1`:

```
OnRadioRxDone: payload = 200017e4, size = 100, rssi = 0, snr = 8
[200017e4]  eb 00 e1 d5   |....|
[200017e8]  00 64 00 00   |.d..|
[200017ec]  56 00 00 82   |V...|
[200017f0]  40 36 80 00   |@6..|
[200017f4]  30 6a dd 50   |0j.P|
[200017f8]  40 8e ff ff   |@...|
[200017fc]  8f d0 60 00   |..`.|
[20001800]  79 7f f0 00   |y...|
[20001804]  10 40 17 f9   |.@..|
[20001808]  de 00 35 40   |..5@|
[2000180c]  00 00 a7 a6   |....|
[20001810]  00 e5 40 20   |..@ |
[20001814]  88 90 54 7f   |..T.|
[20001818]  00 97 00 7f   |....|
[2000181c]  1c 00 dc 0b   |....|
[20001820]  20 d2 83 b9   | ...|
[20001824]  00 2d 00 20   |.-. |
[20001828]  64 03 e8 c2   |d...|
[2000182c]  00 00 fc 92   |....|
[20001830]  f8 ff 00 98   |....|
[20001834]  e1 14 00 00   |....|
[20001838]  a4 a7 ff 40   |...@|
[2000183c]  40 59 80 71   |@Y.q|
[20001840]  00 00 88 22   |..."|
[20001844]  4e 00 00 10   |N...|


OnRadioRxDone: payload = 200017e4, size = 0, rssi = 65438, snr = 30
```

Which results in an 65280-byte out-of-bounds `memcopy` likely with partially controlled attacker data:

LoRaMac-node/src/mac/LoRaMac.c
Lines 1363 to 1364 in a166830

```
1363      case FRAME_TYPE_PROPRIETARY:
1364          memcpy1( MacCtx.RxPayload, &payload[pktHeaderLen], size - pktHeaderLen );
```

LoRaMac-node/src/mac/LoRaMac.c
Lines 127 to 130 in a166830

```
127          /*
128          * Buffer containing the upper layer data.
```

```
129         */
130         uint8_t RxPayload[LORAMAC_PHY_MAXPAYLOAD];
```

LoRaMac-node/src/mac/LoRaMac.c
Lines 51 to 54 in a166830

```
51      /*!
52       * Maximum PHY layer payload size
53       */
54      #define LORAMAC_PHY_MAXPAYLOAD                      255
```

Impact

- Corrupting a large part if the data section is likely to cause a DoS.
- If the large out-of-bounds write does not immediately crash the attacker may gain control over the execution due to now controlling large parts of the data section. (RCE)

Patches

Commit `e851b07` fixes this vulnerability and is available on master and v5.0.0-branch branches.
Will be released with v4.7.0 version

Workarounds

Patch earlier versions with changes provided by commit `e851b07`

References

N/A

For more information

If you have any questions or comments about this advisory:

- Open an issue in https://github.com/Lora-net/LoRaMac-node/issues
- Email us at LoRa-Net@semtech.com

Severity

Moderate

CVE ID

CVE-2022-39274

Weaknesses

No CWEs

Credits

 **SWW13**