CVE-2022-30904

Form-Submission Information

[Suggested description]
In Bestechnic Bluetooth Mesh SDK (BES2300) V1.0, a buffer overflow vulnerability can be triggered during provisioning, because there is no check for the SegN field of the Transaction Start PDU.

[Vulnerability Type]
Buffer Overflow

[Vendor of Product]
Bestechnic : http://www.bestechnic.com/

[Affected Product Code Base]
Bestechnic Bluetooth Mesh SDK (BES2300) - V1.0

[Affected Component]
Affected source code file is prov.c
Affected function is gen_prov_start()

[Attack Type]
Remote

[Impact Code execution]
true

[Impact Denial of Service]
true

[Attack Vectors]
This vulnerability can be triggered during Bluetooth Mesh provisioning. The attack vector is sending malformed segmented packets to victim, without user interaction.

[Has vendor confirmed or acknowledged the vulnerability?]
true

[Discoverer]
Han Yan, Lewei Qu, Dongxiang Ke of Baidu AIoT Security Team

Vulnerability description

In Bestechnic bluetooth mesh core stack, an out-of-bound write vulnerability can be triggered during provisioning, for lacking check for *SegN* in Transaction Start PDU. The vulnerable function is *gen_prov_start* in *prov.c*.

Vulnerability analysis

Analysis

In *gen_prov_start*, there is no check for *SegN* performed. It considers an oversized *SegN* (i.e., greater than 2, corresponding to buffer size 65) as valid, and save it to *link.rx.last_seg*.

```
BT_DBG("len %u last_seg %u total_len %u fcs 0x%02x", buf->len,
       START_LAST_SEG(rx->gpc), link.rx.buf->len, link.rx.fcs);

if (link.rx.buf->len < 1)
{
    BT_ERR("Ignoring zero-length provisioning PDU");
    close_link(PROV_ERR_NVAL_FMT, CLOSE_REASON_FAILED);
    return;
}

if (link.rx.buf->len > link.rx.buf->size)
{
    BT_ERR("Too large provisioning PDU (%u bytes)",
           link.rx.buf->len);
    close_link(PROV_ERR_NVAL_FMT, CLOSE_REASON_FAILED);
    return;
}

if (START_LAST_SEG(rx->gpc) > 0 && link.rx.buf->len <= 20)
{
    BT_ERR("Too small total length for multi-segment PDU");
    close_link(PROV_ERR_NVAL_FMT, CLOSE_REASON_FAILED);
    return;
}

link.rx.seg = (1 << (START_LAST_SEG(rx->gpc) + 1)) - 1;
link.rx.last_seg = START_LAST_SEG(rx->gpc);
memcpy(link.rx.buf->data, buf->data, buf->len);
XACT_SEG_RECV(0);
```

By sending malformed Transaction Start PDU with legal *TotalLength* and oversize *SegN*, the check *SegO* > *SegN* in Transaction Continue PDU can be bypassed.

```
if (seg > link.rx.last_seg)
{
    BT_ERR("Invalid segment index %u", seg);
    close_link(PROV_ERR_NVAL_FMT, CLOSE_REASON_FAILED);
    return;
}
```

In consequence, a Transaction Continue PDU with an actually oversized *SegO* (i.e., greater than 2) will trigger out-of-bound write.