Nordic Bluetooth Mesh SDK transport reassemble-heap overflow 1

Thanks for reviewing !

Any question please contact us at jlu2014yanhan@163.com

Vulnerability description

Nordic Semiconductor is a fabless semiconductor company specializing in wireless technology for the IoT.
Official website : https://www.nordicsemi.com/

In Nordic nRF5 SDK for Mesh, a heap overflow vulnerability can be triggered by sending a series of segmented packets with *SegO* > *SegN*.

The affected SDK is nRF5 SDK for Mesh. https://www.nordicsemi.com/Products/Development-software/nRF5-SDK-for-Mesh/Download?lang=en#infotabs
The affected version is : version <= v5.0.0
The vulnerable function is *trs_seg_packet_in* in *mesh/core/src/transport.c*.

Vulnerability analysis

Analysis
SegO is a lower tansport layer field that indicates the segment offset number.
SegN is a lower transport layer field that indicates the last segment number.

When received first segmented packet, the mesh sdk will allocate a heap buffer to cache the remaining segmented packets:

the length of buffer is (*SegN* + 1)×*single_pdu_size*, where *single_pdu_size* is 8 or 12, depending on *CTL*:

The mesh sdk then continues to receive the remaining segmented packets, copies them into the allocated buffer, where the destination address of *memcpy* is:

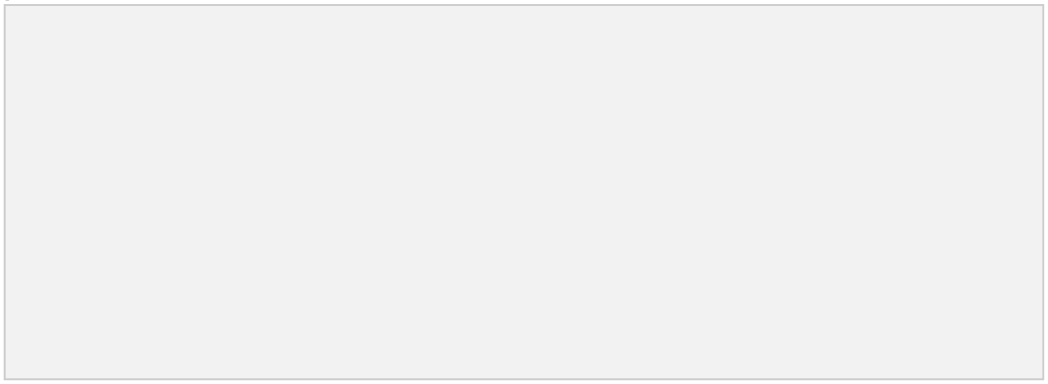
pbuffer + *SegO* * *single_pdu_size*

The mesh sdk doesn't check whether *SegO* <= *SegN* when caching packets. if *SegO* of currently received packet is greater than *SegN* of firstly received packet, a heap overflow will occur.

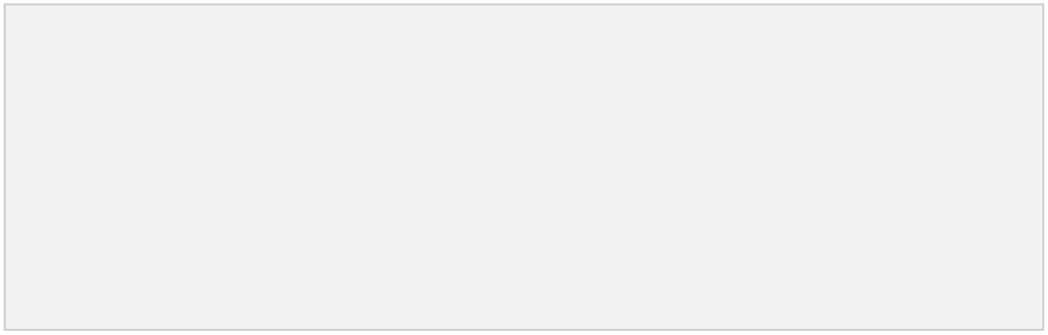
POC

First, we send an access packet with *SegN* 1. The mesh sdk allocates a 24 bytes buffer to cache the remaining packets.

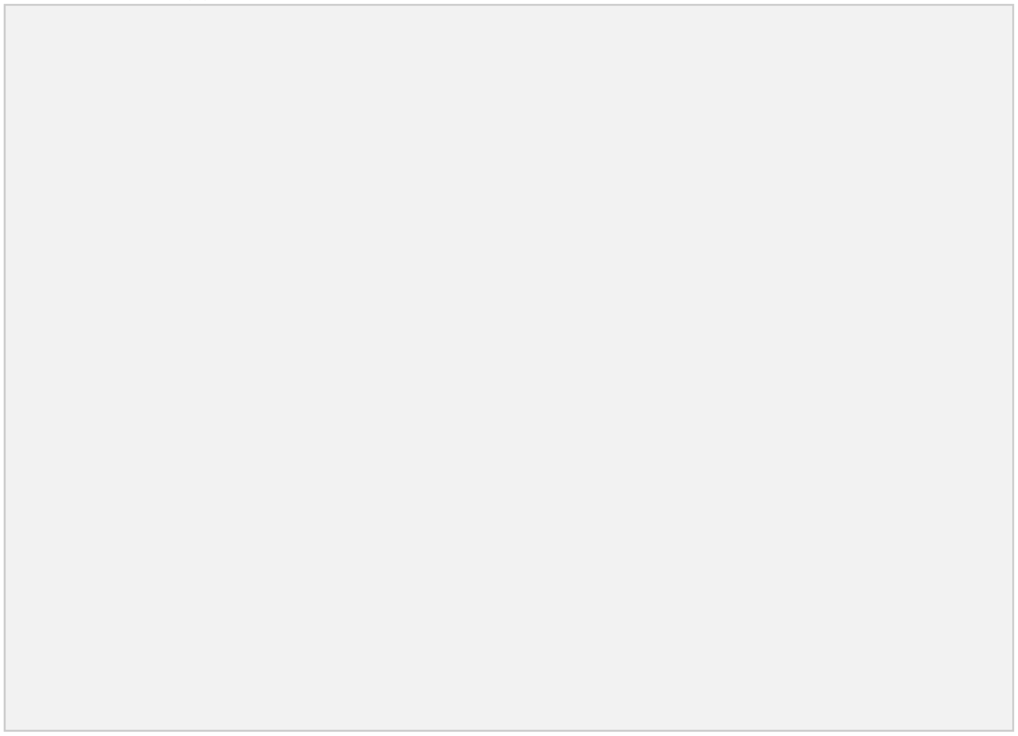
We then send an access packet with *SegN* 1 and *SegO* 2. Since *SeqZero* is the same, this packet will be cached into the previously allocated buffer. However, since the *SegO* is 2, the segment data will be copied into buffer[24] ~ buffer[35], causing a heap overflow. Similarly, we can also send other packet with *SegO* greater than 1 to write to other area.



We added log print before *mesh_mem_alloc* in the *sar_ctx_alloc* and *memcpy* in the *trs_seg_packet_in*. The log demonstrates that allocated buffer size is 24, while the segment offset can be greater than 24, causing heap overflow.



SEGGER Debugger shows the memory state of heap overflow.



References

Bluetooth Mesh : https://www.bluetooth.com/blog/introducing-bluetooth-mesh-networking/
Bluetooth Mesh Profile : https://www.bluetooth.com/specifications/specs/mesh-profile-1-0-1/