Potential Vulnerabilities in the Contiki-ng Stack

Out-of-Bounds Read in SNMP when decoding string

Description

The function snmp_ber_decode_string_len_buffer in os/net/app-layer/snmp/snmp-ber.c (line 359 as of 9e8bb44) decodes string length from a received SNMP packet. At one point, it lacks a validation check on the remaining packet size. This can lead to dereferencing and reading from an out-of-bounds pointer. Our analysis indicates that at most one additional byte can be read beyond the limit of the array.

Technical Details

The snmp-ber functions that decode fields from a received SNMP packet commonly verify that data remains in the packet (*snmp_packet->used != 0*) before reading from the packet. However, the pointer dereferencing in line 375 in the snmp-ber.c file lacks this check.

Hence, if the snmp_ber_decode_string_len_buffer function is called with only one byte left in the SNMP packet, the byte is read as the type field in line 363. Then, line 375 reads out of bound to verify the number of bytes in the length field.

Impact

An out-of-bounds read can cause a crash and lead to a denial of service. This condition could occur on Contiki-ng deployments that support memory safety checks at the hardware level or the software level (e.g., via ASAN-style instrumentation).

Fix Recommendation

We recommend adding the *if(snmp_packet->used == 0)* before line 375 to verify the packet still has at least one byte that can be read. Consequently, the corresponding checks in lines 377 and 385 (the two branches of the if condition) will become redundant and can be removed. We have attached a proposed patch.

Out-of-Bounds Read in SNMP message decoding

Description

The function snmp_message_decode in os/net/app-layer/snmp/snmp-message.c (line 189 as of 9e8bb44) lacks a validation check on the packet size and can dereference and read from an out-of-bound pointer. Our analysis indicates at most one byte can be read past the limit of the array.

Technical Details

This vulnerability exists in the snmp_message_decode function in the snmp-message.c file and is similar to the previous vulnerability reported in the snmp_ber_decode_string_len_buffer function.

After successfully decoding oid data from the received SNMP packet (in line 315), the affected function does not verify the availability of data in the packet before dereferencing and reading from the snmp_packet-in pointer in line 320.

Hence, if the OID data is the last in the packet (as shown by the green bytes in the sample SNMP packet below), the snmp_message_decode function will read out-of-bounds after processing these packets.

A sample SNMP packet that could trigger this:

```
0x30,0x46,0x02,0x01,0x01,0x04,0x0a,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x30,0xa0,0x35,0x2,0x04,0x30,0x35,0x3a,0x34,0x02,0x04,0x30,0x37,0x3a,0x36,0x02,0x01,0x00,0x30,0x24,0x30,0x14,0x06,0x0f,0x81,0xdd,0xec,0xb6,0x01,0xff,0x7d,0x8c,0xeb,0xf9,0x8e,0x10,0x81,0xff,0x7d
```

Impact

Similar to the first vulnerability, an out-of-bounds read can cause a crash and lead to a denial of service. This condition could occur on Contiki-ng deployments that support memory safety checks at the hardware level or are using ASAN-style instrumentation.

Fix Recommendation

We recommend adding the *if(snmp_packet->used == 0)* validation before line 320 to verify the packet still has at least one byte that can be read.