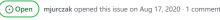
New issue

Stack overflow in SNMP bulk request processing #1353



Labels bug/vulnerability

<> Code ⊙ Issues 179 \$\(\) Pull requests 58 □ Discussions ⊙ Actions □ Wiki

mjurczak commented on Aug 17, 2020

Contributor

Jump to bottom

Description of defect

References:

https://github.com/contiki-ng/contiki-ng/tree/release/v4.5 https://github.com/contiki-ng/contiki-ng/tree/release/v4.4

File

snmp-engine.c

snmp-message.c

Analysis:

Memory access out of buffer boundaries may occur if an SNMP bulk get request with number of OIDs larger than supported by the engine is received and processed.

The OIDs listed in a request are processed by snmp_message_decode() function without verification of the varbinds buffer capacity. The varbinds memory buffer is written with the values provided in SNMP request:

```
contiki-ng/os/net/app-layer/snmp/snmp-message.c
Line 245 in 23db957

245 buf = snmp_oid_decode_oid(buf, &buf_len, varbinds[i].oid, &oid_len);
```

The buffer capacity is determined by:

```
contiki-ng/os/net/app-layer/snmp/snmp-confh
Lines 81 to 87 in 23dob57

81 #define SNMP_MAX_NR_VALUES SNMP_CONF_MAX_NR_VALUES
82 #else
83 /**
84 * \brief Default maximum number of OIDs in one response
85 */
86 #define SNMP_MAX_NR_VALUES 2
87 #endif
```

SNMP get bulk requests are processed by snmp_engine_get_bulk() function that allocates a local stack buffer for buffering OIDs of the requested variables.

The stack buffer in snmp_engine_get_bulk() is populated with OIDs as a first step before any further processing of the data.

The varbinds_length variable value is not verified against the capacity of the temporary oid stack buffer. If the number of requested OIDs exceeds the buffer capacity a stack buffer overflow condition occurs and stack memory beyond the allocated oid buffer is overwritten with OIDs received in SNMP get bulk request.

As the OIDs are supplied in the request content it may be possible to alter the return address from the snmp_engine_get_bulk() function. If the target architecture uses common addressing space for program and data memory (which is common in IoT devices) it may also be possible to supply code in the SNMP request payload and redirect the execution path to the injected code by modification of the return address.

Type:

- Out-of-bounds memory write
- Stack memory overwrite
- Return address alteration
- Remote altering of code execution path
- Remote executable code injection
- Remote code execution

Result

Target(s) affected by this defect ?	
• contiki-ng v4.5	
• contiki-ng v4.4	
Fix	
Rudimentary fix to address the most critical aspect of the issue:	
https://github.com/mjurczak/contiki-ng/tree/bugfix/snmp-engine	
How is this defect reproduced ?	
An example SNMP request causing stack overwrite:	
306102010104067075626C6963A554020431D065A702010402010A3046300C06082B060102010102000500300C06082B060102010102010500300C06082B0601 02010102020500300C06082B060102010102030500300C06082B060102010102040500	
ÇZ ← mjurczak mentioned this issue on Aug 17, 2020	
Bugfix/snmp engine #1355	
(1- Merged)	
ÇZ § Yagoor mentioned this issue on Sep 8, 2020	
SNMP Engine - New Unit Tests #1376	
(⊙Closed)	
g-oikonomou commented on Nov 25, 2020	Member
@Yagoor @miurrzak: Am Lright to assume that this has been fixed in #1355 and/or #1397? Can we close?	
@Yagoor @mjurczak: Am I right to assume that this has been fixed in #1355 and/or #1397? Can we close?	
@Yagoor @mjurczak: Am I right to assume that this has been fixed in #1355 and/or #1397? Can we close?	
@Yagoor @mjurczak: Am I right to assume that this has been fixed in #1355 and/or #1397? Can we close? © g-oikonomou added the bug/vulnerability label on Nov 25, 2020	
© g-oikonomou added the bug/vulnerability label on Nov 25, 2020	
© g-oikonomou added the bug/vulnerability label on Nov 25, 2020 Assignees	
© g-oikonomou added the bug/vulnerability label on Nov 25, 2020 Assignees	
© g-oikonomou added the bug/vulnerability label on Nov 25, 2020 Assignees No one assigned	
© g-oikonomou added the bug/vulnerability label on Nov 25, 2020 Assignees No one assigned Labels	
© g-oikonomou added the bug/vulnerability label on Nov 25, 2020 Assignees No one assigned Labels bug/vulnerability	
© g-oikonomou added the bug/vulnerability label on Nov 25, 2020 Assignees No one assigned Labels bug/vulnerability Projects	
© g-oikonomou added the bug/vulnerability label on Nov 25, 2020 Assignees No one assigned Labels bug/vulnerability	
Assignees No one assigned Labels bug/vulnerability Projects None yet	
Assignees No one assigned Labels bug/vulnerability Projects None yet	
Assignees No one assigned Labels bug/vulnerability Projects None yet	
Assignees No one assigned Labels bug/vulnerability Projects None yet Milestone No milestone No milestone	
Assignees No one assigned Labels bug/vulnerability Projects None yet Milestone No milestone No milestone Development	
Assignees No one assigned Labels bug/vulnerability Projects None yet Milestone No milestone No milestone	
Assignees No one assigned Labels bug/vulnerability Projects None yet Milestone No milestone No milestone Development	
Assignees No one assigned Labels bug/vulnerability Projects None yet Milestone No milestone Development No branches or pull requests	

Memory corruption