

obgm / libcoap Public

<> Code Issues 52 Pull requests 7 Actions Projects Wiki Security

Commit b7847c4



mrdeep1 committed last month

sanitizer: Fix reported issues

coap_new_cache_entry() does not correctly check for no PDU data when called with COAP_CACHE_RECORD_PDU. No current libcoap code (examples and library) call coap_new_cache_entry() with COAP_CACHE_RECORD_PDU set.

Internal function coap_pdu_resize() can be used to reduce a PDU size, creating current options confusion. Fix is not to reduce PDU if new size is smaller than the current used size. No current libcoap code calls coap_pdu_resize() to reduce the size.

If there is an issue with the PDU options where the maximum used option value is larger than the last defined option value, an assert() is triggered.

All of the coap_*_option() functions correctly manage pdu->max_opt, but this issue could occur if coap_pdu_resize() was called to reduce the PDU size below that of pdu->used_size.

release-4.3.5-patches · v4.3.5b

1 parent 37ee78a commit b7847c4

2 files changed +11 -3 lines changed

↑ Top ⚙️

Filter files...

src

coap_cache.c

coap_pdu.c

2 files changed +11 -3 lines changed

Search within code ⚙️

src/coap_cache.c

```

↑
@@ -203,7 +203,8 @@ coap_new_cache_entry_lkd(coap_session_t *session, const
coap_pdu_t *pdu,
203 203     memcpy(entry->pdu, pdu, offsetof(coap_pdu_t, token));
204 204     memcpy(entry->pdu->token, pdu->token, pdu->used_size);
205 205     /* And adjust all the pointers etc. */
206 -     entry->pdu->data = entry->pdu->token + (pdu->data - pdu->token);
206 +     if (pdu->data)
207 +         entry->pdu->data = entry->pdu->token + (pdu->data - pdu->token);
207 208     }
208 209     }
209 210     entry->cache_key = coap_cache_derive_key(session, pdu, session_based);
↓

```

```

src/coap_pdu.c
↑
@@ -280,10 +280,12 @@ coap_pdu_duplicate_lkd(const coap_pdu_t *old_pdu,
280 280     int
281 281     coap_pdu_resize(coap_pdu_t *pdu, size_t new_size) {
282 282         if (new_size > pdu->alloc_size) {
283 +         /* Expanding the PDU usage */
283 284         #if !defined(WITH_LWIP)
284 285             uint8_t *new_hdr;
285 286             size_t offset;
286 287         #endif
288 +
287 289         if (pdu->max_size && new_size > pdu->max_size) {
288 290             coap_log_warn("coap_pdu_resize: pdu too big\n");
289 291             return 0;
↓
↑
@@ -314,8 +316,8 @@ coap_pdu_resize(coap_pdu_t *pdu, size_t new_size) {
314 316         else
315 317             pdu->actual_token.s = &pdu->token[2];
316 318         #endif
319 +         pdu->alloc_size = new_size;
317 320     }
318 -         pdu->alloc_size = new_size;
319 321         return 1;
320 322     }
321 323
↓
↑
@@ -629,7 +631,12 @@ coap_insert_option(coap_pdu_t *pdu, coap_option_num_t
number, size_t len,

```

```
629 631     }
630 632     prev_number = opt_iter.number;
631 633     }
632 -   assert(option != NULL);
634 +   if (option == NULL) {
635 +     /* Code is broken somewhere */
636 +     coap_log_warn("coap_insert_option: Broken max_opt\n");
637 +     return 0;
638 +   }
639 +
633 640     /* size of option inc header to insert */
634 641     shift = coap_opt_encode_size(number - prev_number, len);
635 642
```



Comments 0



Please [sign in](#) to comment.