Commit beee2dc

sbabic committed on Mar 24

```
mongoose: Integer Underflow in Multipart Upload Parser

The function mg_http_multipart_continue_wait_for_chunk() has
a discrepancy between its guard condition and a subsequent
subtraction in the else branch.  The guard at line 250 checks
`(int) io->len < mp_stream->boundary.len + 6`, allowing execution
to continue when io->len >= boundary.len + 6.
However, when mg_strstr() finds the boundary string in the
buffer (else branch at line 264), data_len is computed as
`io->len - (mp_stream->boundary.len + 8)`. The +6 vs +8
mismatch means that when io->len is in the range [boundary.len + 6,
boundary.len + 7], the subtraction underflows the size_t
variable to SIZE_MAX or SIZE_MAX - 1.

This will fix CVE-2026-28525.

Description of issue copied from vulnerability report - many thanks to
Kazuma for his analyses.

Signed-off-by: Stefano Babic <stefano.babic@swupdate.org>
Reported by: Kazuma Matsumoto, a security researcher at GMO Cybersecurity by IERAE, Inc."
```

master

1 parent e3b3c97 commit beee2dc

1 file changed +3 -3

Top

Filter files...

mongoose

mongoose_multipart.c

Search within code

mongoose/mongoose_multipart.c

```
@@ -261,12 +261,12 @@ static int
mg_http_multipart_continue_wait_for_chunk(struct mg_connection *c) {
261 261           }
262 262           return 0;
263 263         } else {
264     -         size_t data_len = io->len - (mp_stream->boundary.len + 8);
    264 +         size_t data_len = io->len - (mp_stream->boundary.len + 6);
265 265           size_t consumed = mg_http_multipart_call_handler(c,
MG_EV_HTTP_PART_DATA,
266     -                                                          (char *) io->buf,
data_len);
    266 +                                                           (char *) io->buf, data_len);
267 267           mg_iobuf_del(io, 0, consumed);
268 268           if (consumed == data_len) {
269     -             mg_iobuf_del(io, 0, mp_stream->boundary.len + 8);
    269 +             mg_iobuf_del(io, 0, mp_stream->boundary.len + 6);
270 270               mp_stream->state = MPS_FINALIZE;
271 271               return 1;
272 272           } else {
```

Comments 0

Please sign in to comment.