New issue

Jump to bottom

Out-of-bounds write caused by incorrect error handling of calloc in mg_tls_init (mongoose.c:3297) #1203

Closed **cve-reporting** opened this issue on Jan 23, 2021 · 1 comment

cve-reporting commented on Jan 23, 2021

Mongoose HTTPS server (compiled with mbedTLS support) is vulnerable to remote OOB write attack via connection request after exhausting memory pool.

Incorrect handling of the value returned by calloc in mg_tls_init may lead to:

- out-of-bound write attempt and segmentation fault error in case of restrictive memory protection,
- near NULL pointer (at 0x458) overwrite in case of limited memory restrictions (e.g. in embedded environments).

Memory allocations are triggered during handling of each HTTPS requests, so the allocation error can be caused remotely by flooding with requests until exhausting the memory.
In some embedded environments near zero memory areas are used to store device configuration, so in this case such configuration can be overwritten remotely.

Vulnerable code (mongoose.c):

```
3255: struct mg_tls {
3256:   char *cafile;             // CA certificate path
3257:   mbedtls_x509_crt ca;      // Parsed CA certificate
3258:   mbedtls_x509_crt cert;    // Parsed certificate
3259:   mbedtls_ssl_context ssl;  // SSL/TLS context
3260:   mbedtls_ssl_config conf;  // SSL-TLS config
3261:   mbedtls_pk_context pk;    // Private key context
3262: };

3296: int mg_tls_init(struct mg_connection *c, struct mg_tls_opts *opts) {
3297:   struct mg_tls *tls = (struct mg_tls *) calloc(1, sizeof(*tls)); printf("mg_tls_init tls = %p %ld\n", tls, &(tls->ssl));
3298:   int rc = 0;
3299:   LOG(LL_DEBUG, ("%lu Setting TLS, CA: %s, cert: %s, key: %s", c->id,
3300:                  opts->ca == NULL ? "null" : opts->ca,
3301:                   opts->cert == NULL ? "null" : opts->cert,
3302:                   opts->certkey == NULL ? "null" : opts->certkey));
3303:   mbedtls_ssl_init(&tls->ssl);
3304:   mbedtls_ssl_config_init(&tls->conf);
3305:   mbedtls_ssl_conf_dbg(&tls->conf, debug_cb, c);
```

See following recommendations for details (especially the calloc example):
https://wiki.sei.cmu.edu/confluence/display/c/ERR33-C.+Detect+and+handle+standard+library+errors

The issue can be reproduced and tested using ErrorSanitizer (https://gitlab.com/ErrorSanitizer/ErrorSanitizer).

Reproduction steps:

0. Install gdb
1. Download and unpack code of ErrorSanitizer (https://gitlab.com/ErrorSanitizer/ErrorSanitizer)
2. Perform compilation of ErrorSanitizer according to the manual (https://gitlab.com/ErrorSanitizer/ErrorSanitizer#compilation)

   cd ErrorSanitizer; make
3. Set ESAN to the path of ErrorSanitizer directory

   export ESAN=/opt/...
4. Download and unzip attached map temp_2.cur_input
   temp_2.cur_input.zip
5. Install mbedTLS library
6. Download, unzip and compile mongoose example "http-restful-server" with define MBEDTLS_DIR set for mbedTLS directory and debug symbols (-g)
7. Run Mongoose "http-restful-server" example with ErrorSanitizer in gdb using:

   gdb -batch -ex='run' -ex='backtrace' --args env LD_PRELOAD="$ESAN/error_sanitizer_preload.so" ./example temp_2.cur_input
8. Open in the browser following URL (where <MONGOOSE_ADDR> is address of tested Mongoose instance):

   https://<MONGOOSE_ADDR>:8000

You should receive similar output:

```
process 30197 is executing new program: mongoose/examples/http-restful-server/example
2021-01-21 00:00:00  I log.c:18:mg_log_set       Setting log level to 2
2021-01-21 00:00:00  I sock.c:453:mg_listen      1 accepting on https://localhost:8000

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff79970d0 in mbedtls_ssl_init () from /usr/lib/x86_64-linux-gnu/libmbedtls.so.10
#0  0x00007ffff79970d0 in mbedtls_ssl_init () from /usr/lib/x86_64-linux-gnu/libmbedtls.so.10
#1  0x0000555555560fa9 in mg_tls_init (c=0x5555557688c0, opts=0x7fffffffdbf0) at src/tls.c:70
#2  0x000055555556377a in fn (c=0x5555557688c0, ev=4, ev_data=0x0, fn_data=0x0) at main.c:28
#3  0x0000555555557df1 in mg_call (c=0x5555557688c0, ev=4, ev_data=0x0) at src/event.c:9
#4  0x0000555555555fae4 in accept_conn (mgr=0x7ffffffffdd20, lsn=0x5555557686c0) at src/sock.c:393
#5  0x0000555555560484 in mg_mgr_poll (mgr=0x7ffffffffdd20, ms=1000) at src/sock.c:543
#6  0x00005555555638de in main () at main.c:51
```

cpq commented on Jan 26, 2021 Member

Pushed 8e52075

cpq closed this as completed on Jan 26, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants