Out-of-bounds write caused by incorrect error handling of calloc in mg_tls_init (mongoose.c:3443) #1204

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

cve-reporting commented on Jan 23, 2021 • edited

Mongoose HTTPS server (compiled with OpenSSL 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 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):

```
3421: struct mg_tls {
3422:   SSL_CTX *ctx;
3423:   SSL *ssl;
3424: };

3442: int mg_tls_init(struct mg_connection *c, struct mg_tls_opts *opts) {
3443:   struct mg_tls *tls = (struct mg_tls *) calloc(1, sizeof(*tls)); printf("tls = %p %ld\n", tls, (long)(&tls->ctx));
3444:   const char *id = "mongoose";
3445:   static unsigned char s_initialised = 0;
3446:   int rc;
3447:   if (!s_initialised) {
3448:     SSL_library_init();
3449:    s_initialised++;
3450:   }

3455:   tls->ctx = c->is_client ? SSL_CTX_new(SSLv23_client_method())
3456:                           : SSL_CTX_new(SSLv23_server_method());
3457:   if ((tls->ssl = SSL_new(tls->ctx)) == NULL) {
3458:     mg_error(c, "SSL_new");
3459:     goto fail;
3460:   }
```

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 OpenSSL library
6. Download, unzip and compile mongoose example "http-restful-server" with define OPENSSL_DIR set for OpenSSL 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 10544 is executing new program: mongoose/examples/http-restful-server-openssl/example
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
2021-01-21 00:00:00  I sock.c:461:mg_listen      1 accepting on https://localhost:8000

Program received signal SIGSEGV, Segmentation fault.
0x000055555560d6a in mg_tls_init (c=0x555555768780, opts=0x7fffffffdbf0) at src/tls.c:209
209     src/tls.c: No such file or directory.
#0  0x000055555560d6a in mg_tls_init (c=0x555555768780, opts=0x7fffffffdbf0) at src/tls.c:209
#1  0x0000555555563860 in fn (c=0x555555768780, ev=4, ev_data=0x0, fn_data=0x0) at main.c:28
#2  0x0000555555557d11 in mg_call (c=0x555555768780, ev=4, ev_data=0x0) at src/event.c:9
#3  0x000055555555fa1d in accept_conn (mgr=0x7fffffffdd10, lsn=0x555555769500) at src/sock.c:398
#4  0x00005555555603bd in mg_mgr_poll (mgr=0x7fffffffdd10, ms=1000) at src/sock.c:551
#5  0x00005555555639bb in main () at main.c:49
```

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