New issue

Jump to bottom

# Out-of-bounds write caused by incorrect error handling of calloc in mg_http_serve_file (mongoose.c:800) #1201

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

**cve-reporting** commented on Jan 23, 2021 • edited

Mongoose HTTP server is vulnerable to remote OOB write attack via connection request after exhausting memory pool.

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

- NULL pointer dereference 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 HTTP 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):

```
427: struct http_data {
428:     void *old_pfn_data;  // Previous pfn_data
429:     FILE *fp;            // For static file serving
430: };

780: void mg_http_serve_file(struct mg_connection *c, struct mg_http_message *hm,
781:                         const char *path, const char *mime, const char *hdrs) {

800: struct http_data *d = (struct http_data *) calloc(1, sizeof(*d));
801: d->fp = fp;
802: d->old_pfn_data = c->pfn_data;
803: c->pfn = static_cb;
804: c->pfn_data = d;
```

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. Remove hook files from from the ErrorSanitizer/hooks directory APART from hooks_memory.c file:

   find ErrorSanitizer/hooks -name "hooks_[acfost]*[.]c" -delete

3. Comment out the whole contents of hooks/hooks_memory.c file APART from the calloc section to disable hooks for: malloc and realloc.

   /*****************************************************************************/

   /* void* calloc(size_t num, size_t size); */
   typedef void *(*calloc_func_t)(size_t num, size_t size);
   static void *real_calloc(size_t num, size_t size)
   ...
   void *calloc(size_t num, size_t size)
   ...

   /*****************************************************************************/

4. Continue with compilation of ErrorSanitizer according to the manual (https://gitlab.com/ErrorSanitizer/ErrorSanitizer#compilation)

   cd ErrorSanitizer; make

5. Set ESAN to the path of ErrorSanitizer directory

   export ESAN=/opt/...

6. Download and unzip attached map temp_3.cur_input
   temp_3.cur_input.zip

7. Download, unzip and compile mongoose example "complete" with debug symbols (-g)

8. Run Mongoose "complete" example with ErrorSanitizer in gdb using:

   gdb -batch -ex='run' -ex='backtrace' --args env LD_PRELOAD="$ESAN/error_sanitizer_preload.so" ./example temp_3.cur_input

9. Open in the browser following URL (where <MONGOOSE_IP> is address of tested Mongoose instance):

   http://<MONGOOSE_IP>:8000/#/logs

(Because memory operations can occur in a different sequence, actions 9. and 10. sometimes need to be executed multiple times.)

You should receive similar output:

```
process 21111 is executing new program: mongoose/examples/complete/example
2021-01-21 00:00:00  I mongoose.c:2899:mg_listen 1 accepting on http://localhost:8000

Program received signal SIGSEGV, Segmentation fault.
0x0000555555558ef1 in mg_http_serve_file (c=0x55555576a8c0, hm=0x7fffffffdb90, path=0x5555555667d4 "log.txt", mime=0x5555555667c9 "text/plain", hdrs=0x5555555667c8 "") at
```

```
../../mongoose.c:800
800           d->fp = fp;
#0  0x0000555555558ef1 in mg_http_serve_file (c=0x55555576a8c0, hm=0x7fffffffdb90, path=0x5555555667d4 "log.txt", mime=0x5555555667c9 "text/plain", hdrs=0x5555555667c8 "") at
../../mongoose.c:800
#1  0x000055555556269e in cb (c=0x55555576a8c0, ev=8, ev_data=0x7fffffffdb90, fn_data=0x7fffffffe250) at main.c:122
#2  0x0000555555557581 in mg_call (c=0x55555576a8c0, ev=8, ev_data=0x7fffffffdb90) at ../../mongoose.c:397
#3  0x00005555555598f5 in http_cb (c=0x55555576a8c0, ev=5, ev_data=0x7fffffffe170, fn_data=0x7fffffffe250) at ../../mongoose.c:1146
#4  0x0000555555557557 in mg_call (c=0x55555576a8c0, ev=5, ev_data=0x7fffffffe170) at ../../mongoose.c:396
#5  0x000055555555e30e in read_conn (c=0x55555576a8c0, fn=0x55555555dc3a <ll_read>) at ../../mongoose.c:2689
#6  0x000055555555f71a in mg_mgr_poll (mgr=0x7fffffffe250, ms=100) at ../../mongoose.c:2995
#7  0x0000555555562af3 in main () at main.c:203
```

**cve-reporting** changed the title ~~NULL pointer dereference caused by incorrect error handling of calloc in mg_http_serve_file (mongoose.c:800)~~ **Out-of-bounds write caused by incorrect error handling of calloc in mg_http_serve_file (mongoose.c:800)** on Jan 23, 2021

**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