ezXML Bugs

Status: Beta
Brought to you by: voisine

#22 Out-of-bounds write caused by incorrect error handling of malloc in ezxml_new (ezxml.c:843)

Milestone: v1.0 (example) **Status:** open **Owner:** Aaron Voisine **Labels:** None
Priority: 5
Updated: 2021-10-25 **Created:** 2021-01-24 **Creator:** CVE Reporting **Private:** No

ezxml is vulnerable to OOB write when opening XML file after exhausting the memory pool.

Incorrect handling of the value returned by malloc in ezxml_new 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 opening XML files, so the allocation error can be caused locally or remotely depending on the way of obtaining files.
In some embedded environments near zero memory areas are used to store device configuration, so in this case such configuration can be overwritten using this vulnerability.

Vulnerable code (ezxml.c):

```
833: // returns a new empty ezxml structure with the given root tag name
834: ezxml_t ezxml_new(const char *name)
835: {
836:     static char *ent[] = { "lt;", "&#60;", "gt;", "&#62;", "quot;", "&#34;",
837:                            "apos;", "&#39;", "amp;", "&#38;", NULL };
...
843:     root->ent = memcpy(malloc(sizeof(ent)), ent, sizeof(ent));
```

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:

1. Install gdb
2. Download and unpack code of ErrorSanitizer (https://gitlab.com/ErrorSanitizer/ErrorSanitizer)
3. Perform compilation of ErrorSanitizer according to the manual (https://gitlab.com/ErrorSanitizer/ErrorSanitizer#compilation)
   cd ErrorSanitizer; make
4. Set ESAN to the path of ErrorSanitizer directory
   export ESAN=/opt/...
5. Download attached map temp_1.cur_input
6. Download and compile ezxml 0.8.6
7. Run ezxml test program example with ErrorSanitizer in gdb using:
   gdb -batch -ex='run' -ex='backtrace' -ex='backtrace full' --args env LD_PRELOAD=$ESAN/error_sanitizer_preload.so ./ezxmltest temp_1.cur_input

You should receive similar output:

```
Program received signal SIGSEGV, Segmentation fault.
0x00005555555599e0 in ezxml_new (name=0x0) at ezxml.c:843
843     root->ent = memcpy(malloc(sizeof(ent)), ent, sizeof(ent));
#0  0x00005555555599e0 in ezxml_new (name=0x0) at ezxml.c:843
#1  0x000055555555756d in ezxml_parse_str (s=0x7ffff7ff5000 "<TAG1>VALUE</TAG1>\n", len=19)
#2  0x00005555555584c4 in ezxml_parse_fd (fd=3) at ezxml.c:641
#3  0x00005555555585c4 in ezxml_parse_file (file=0x7fffffffe222 "temp_1.esn_input") at ezxml
#4  0x000055555555a53a in main (argc=2, argv=0x7fffffffde78) at ezxml.c:1008
#0  0x00005555555599e0 in ezxml_new (name=0x0) at ezxml.c:843
    ent = {0x55555555a97e "lt;", 0x55555555a982 "&#60;", 0x55555555a988 "gt;", 0x55555555a9
    root = 0x555555761950
#1  0x000055555555756d in ezxml_parse_str (s=0x7ffff7ff5000 "<TAG1>VALUE</TAG1>\n", len=19)
    root = 0x555555554f80 <_start>
    q = 0 '\000'
    e = 0 '\000'
    d = 0x5400000054 <error: Cannot access memory at address 0x5400000054>
    attr = 0x1012
    a = 0x7fffffffdd20
    l = 64
    i = 0
    j = 84
#2  0x00005555555584c4 in ezxml_parse_fd (fd=3) at ezxml.c:641
    root = 0x0
    st = {st_dev = 66311, st_ino = 2527224, st_nlink = 1, st_mode = 33188, st_uid = 1000, st
    l = 4096
    m = 0x7ffff7ff5000
#3  0x00005555555585c4 in ezxml_parse_file (file=0x7fffffffe222 "temp_1.esn_input") at ezxml
    fd = 3
    xml = 0x7ffff7db3c79 <line+25>
#4  0x000055555555a53a in main (argc=2, argv=0x7fffffffde78) at ezxml.c:1008
    xml = 0x7fffffffde70
    s = 0x0
    i = 21845
```

1 Attachments

temp_1.cur_input

Discussion

Egbert Eich - *2021-10-25*

The proposed patch addresses the issue demonstrated by the attached test case.
All said in this comment applies.

Last edit: Egbert Eich 2021-10-25

Fix-CVE-2021-26222-bug-22.patch

Log in to post a comment.