ezXML Bugs

Status: Beta
Brought to you by: voisine

#23 Out-of-bounds write caused by incorrect error handling of malloc in ezxml_new (ezxml.c:750)

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_toxml 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.

Declaration (ezxml.h):

```
37: #define EZXML_BUFSIZE 1024 // size of internal memory buffers
```

Vulnerable code (ezxml.c):

```
743: // Converts an ezxml structure back to xml. Returns a string of xml data that
744: // must be freed.
745: char *ezxml_toxml(ezxml_t xml)
746: {
747:     ezxml_t p = (xml) ? xml->parent : NULL, o = (xml) ? xml->ordered : NULL;
748:     ezxml_root_t root = (ezxml_root_t)xml;
749:    size_t len = 0, max = EZXML_BUFSIZE;
750:     char *s = strcpy(malloc(max), ""), *t, *n;
```

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_2.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_2.cur_input

You should receive similar output:

```
process 10454 is executing new program: ezxml/ezxmltest

Program received signal SIGSEGV, Segmentation fault.
0x00005555555590ce in ezxml_toxml (xml=0x555555761950) at ezxml.c:750
750     char *s = strcpy(malloc(max), ""), *t, *n;
#0  0x00005555555590ce in ezxml_toxml (xml=0x555555761950) at ezxml.c:750
#1  0x000055555555a54a in main (argc=2, argv=0x7fffffffde78) at ezxml.c:1009
#0  0x00005555555590ce in ezxml_toxml (xml=0x555555761950) at ezxml.c:750
    p = 0x0
    o = 0x0
    root = 0x555555761950
    len = 0
    max = 1024
    s = 0x979946078df8ca00 <error: Cannot access memory at address 0x979946078df8ca00>
    t = 0x1 <error: Cannot access memory at address 0x1>
    n = 0x5555555585d8 <ezxml_parse_file+72> "H\....\002"
    i = 0
    j = 0
    k = 0
#1  0x000055555555a54a in main (argc=2, argv=0x7fffffffde78) at ezxml.c:1009
    xml = 0x555555761950
    s = 0x0
    i = 21845
```

1 Attachments

temp_2.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.

Fix-CVE-2021-26220-bug-23.patch

Log in to post a comment.

SourceForge

Create a Project

Open Source Software

Business Software

Top Downloaded Projects

Company

About

Team

SourceForge Headquarters

225 Broadway Suite 1600

San Diego, CA 92101

+1 (858) 454-5900

Resources

Support

Site Documentation

Site Status

Terms Privacy Opt Out Advertise