New issue

Jump to bottom

NullPointer in actions.c:701:7 #4

Closed **NigelX** opened this issue on Feb 25, 2021 · 4 comments

Assignees

NigelX commented on Feb 25, 2021 • edited

Project: exif
system: ubuntu 20.04
Fuzzer: afl_exif_out_xml
poc.zip

Command:

```
./exif poc.jpeg -x
```

asan

```
AddressSanitizer:DEADLYSIGNAL
=================================================================
==3675403==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000000 (pc 0x0000004cd901 bp 0x7ffd419c2ce0 sp 0x7ffd419c2300 T0)
==3675403==The signal is caused by a WRITE memory access.
==3675403==Hint: address points to the zero page.
    #0 0x4cd901 in escape_xml /home/hx_server/target/libexif/ASAN_exif/exif/actions.c:701:7
    #1 0x4cd48a in show_entry_xml /home/hx_server/target/libexif/ASAN_exif/exif/actions.c:723:26
    #2 0x7f0f5af7fedd in exif_content_foreach_entry /home/hx_server/target/libexif/libexif/exif-content.c:225:3
    #3 0x7f0f5af85b71 in exif_data_foreach_content /home/hx_server/target/libexif/libexif/exif-data.c:1168:3
    #4 0x4cc512 in action_tag_list_xml /home/hx_server/target/libexif/ASAN_exif/exif/actions.c:745:3
    #5 0x4cf155 in main /home/hx_server/target/libexif/ASAN_exif/exif/main.c:474:4
    #6 0x7f0f5ad430b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/csu/../csu/libc-start.c:308:16
    #7 0x41d7cd in _start (/home/hx_server/target/libexif/ASAN_exif/exif/exif+0x41d7cd)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV /home/hx_server/target/libexif/ASAN_exif/exif/actions.c:701:7 in escape_xml
==3675403==ABORTING
```

gdb

```
Program received signal SIGSEGV, Segmentation fault.
0x00000000004cd901 in escape_xml (text=<optimized out>) at actions.c:701
701             *out = '\x0';  /* NUL terminate the string */

[ Legend: Modified register | Code | Heap | Stack | String ]
─────────────────────────────────────────────────────────────────────────────────────────────────────────────── registers ────
$rax   : 0x0
$rbx   : 0x0
$rcx   : 0x40
$rdx   : 0x0
$rsp   : 0x00007fffffffcc20  →  0x00007fffffffd560  →  0x0000000000000000
$rbp   : 0x00007fffffffd600  →  0x00007fffffffd760  →  0x00007fffffffe240  →  0x0000000000000000
$rsi   : 0x0000603000000340  →  0x34030000dc000000
$rdi   : 0x00007fffffffcc60  →  0x0000000000000000
$rip   : 0x00000000004cd901  →  <escape_xml+737> mov BYTE PTR [rbx], 0x0
$r8    : 0x11
$r9    : 0xffffefff80008678
$r10   : 0x00007ffff7df7e56  →  0x007970636e727473 ("strncpy"?)
$r11   : 0x0000000000482650  →  <strncpy+0> push rbp
$r12   : 0x00007fffffffd0e0  →  "Manufacturer"
$r13   : 0x0000000000515888  →  0x00007ffff7d916a0  →  0x00000000fbad2a84  →  0x0000000000000000
$r14   : 0x00007fffffffcc60  →  0x0000000000000000
$r15   : 0x00000fffffffff988  →  0x0000000000000000
$eflags: [ZERO carry PARITY adjust sign trap INTERRUPT direction overflow RESUME virtualx86 identification]
$cs: 0x0033 $ss: 0x002b $ds: 0x0000 $es: 0x0000 $fs: 0x0000 $gs: 0x0000
─────────────────────────────────────────────────────────────────────────────────────────────────────────────────── stack ────
0x00007fffffffcc20│+0x0000: 0x00007fffffffd560  →  0x0000000000000000    ← $rsp
0x00007fffffffcc28│+0x0008: 0x00007ffff7d916a0  →  0x00000000fbad2a84  →  0x0000000000000000
0x00007fffffffcc30│+0x0010: 0x00000fffffffff988  →  0x0000000000000000
0x00007fffffffcc38│+0x0018: 0x00000000004cd48b  →  <show_entry_xml+1131> mov rdi, rax
0x00007fffffffcc40│+0x0020: 0x0000000041b58ab3
0x00007fffffffcc48│+0x0028: 0x00000000004eb372  →  "2 32 1024 5 v:709 1184 1024 5 t:709"
0x00007fffffffcc50│+0x0030: 0x00000000004cd020  →  <show_entry_xml+0> lea rsp, [rsp-0x98]
0x00007fffffffcc58│+0x0038: 0x0000619000000980  →  0x0000000000000000
──────────────────────────────────────────────────────────────────────────────────────────────────────────────── code:x86:64 ────
     0x4cd8f7 <escape_xml+727> mov    al, BYTE PTR [rax+0x7fff8000]
     0x4cd8fd <escape_xml+733> test   al, al
     0x4cd8ff <escape_xml+735> jne    0x4cd950 <escape_xml+816>
 →   0x4cd901 <escape_xml+737> mov    BYTE PTR [rbx], 0x0
     0x4cd904 <escape_xml+740> mov    rax, QWORD PTR [rip+0x8edd15]        # 0xdbb620 <escape_xml.escaped>
     0x4cd90b <escape_xml+747> jmp    0x4cd934 <escape_xml+788>
     0x4cd90d <escape_xml+749> mov    rdi, QWORD PTR [rip+0x8edd0c]        # 0xdbb620 <escape_xml.escaped>
     0x4cd914 <escape_xml+756> call   0x495c10 <free>
     0x4cd919 <escape_xml+761> mov    QWORD PTR [rip+0x8edcfc], 0x0        # 0xdbb620 <escape_xml.escaped>
───────────────────────────────────────────────────────────────────────────────────────────────────── source:actions.c+701 ────
    696                          default:
    697                                  *out = *text;
    698                                  break;
    699                  }
    700          }
 →  701          *out = '\x0';  /* NUL terminate the string */
    702          return escaped;
    703  }
    704
    705  static void
    706  show_entry_xml (ExifEntry *e, void *data)
──────────────────────────────────────────────────────────────────────────────────────────────────────────── threads ────
[#0] Id 1, Name: "exif", stopped 0x4cd901 in escape_xml (), reason: SIGSEGV
────────────────────────────────────────────────────────────────────────────────────────────────────────────── trace ────
[#0] 0x4cd901 → escape_xml(text=<optimized out>)
[#1] 0x4cd48b → show_entry_xml(e=<optimized out>, data=<optimized out>)
[#2] 0x7ffff7e08ede → exif_content_foreach_entry(content=0x603000000160, func=0x4cd020 <show_entry_xml>, data=0x7fffffffd68c)
[#3] 0x7ffff7e0eb72 → exif_data_foreach_content(data=0x6060000021e0, func=0x0, user_data=0x7fffffffd68c)
[#4] 0x4cc513 → action_tag_list_xml(ed=0x7fffffffcc60, p={
  tag = 65535,
  ifd = EXIF_IFD_COUNT,
  machine_readable = 0x0,
  use_ids = 0x0,
  width = 0x50,
  fin = 0x602000000250 "poc.jpeg",
  set_value = 0x0,
  set_thumb = 0x0
})
[#5] 0x4cf156 → main(argc=<optimized out>, argv=<optimized out>)
```

HX from **Topsec alpha Security Team**

msmeissn closed this as completed in `f6334d9` on Feb 25, 2021

msmeissn self-assigned this on Feb 25, 2021

msmeissn commented on Feb 25, 2021 Contributor

i added a null length string check to avoid dereferencing NULL ptr.

msmeissn commented on Apr 12, 2021 Contributor

I currently would not classify this as security issue.

- the crash is caused by a NULL ptr deref, so no other memory corruption
- the output of exif is not corrupted for followup issues
- exif the program terminates anyway, so it is not causing a denial of service.

So I am currently not seeing security impact and would not request a CVE.

carnil commented on Apr 14, 2021

> I currently would not classify this as security issue.
>
> ```
> * the crash is caused by a NULL ptr deref, so no other memory corruption
>
> * the output of exif is not corrupted for followup issues
>
> * exif the program terminates anyway, so it is not causing a denial of service.
> ```
>
> So I am currently not seeing security impact and would not request a CVE.

@msmeissn it looks still someone has requested a CVE, it has https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-27815 (shoult it then be disputed?)

msmeissn commented on Apr 15, 2021 Contributor

I at least filed a description update. I think availability impact can be classified low, so we can leave it valid.

Availability impact: Low
Confidentiality impact: None
Integrity impact: None

Assignees

msmeissn

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants