xmllint: global-buffer-overflow in xmlEncodeEntitiesInternal

Hi, we found a global-buffer-overflow in xmlEncodeEntitiesInternal at libxml2/entities.c:583

version: commit 8e7c20a1 (20910-GITv2.9.10-103-g8e7c20a1)

How to reproduce:

1. build libxml2 with ASAN.
2. run following cmd `xmllint --htmlout $PoC`

🔗 poc

Here's ASAN log.

```
==19600==ERROR: AddressSanitizer: global-buffer-overflow on address 0x5608cd40a230 at pc 0x7f6705c1cdb4 bp 0x7ffced505890 sp 0x7ffce
READ of size 1 at 0x5608cd40a230 thread T0
    #0 0x7f6705c1cdb3 in xmlEncodeEntitiesInternal /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/entities.c
    #1 0x5608cd1ea3f5 in xmlHTMLEncodeSend /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/xmllint.c:531
    #2 0x5608cd1ea3f5 in xmlHTMLError /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/xmllint.c:642
    #3 0x7f6705c7fac9 in __xmlRaiseError /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/error.c:638
    #4 0x7f6705da51ec in xmlFatalErrMsgStrIntStr /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/parser.c:720
    #5 0x7f6705da51ec in xmlParseElementStart /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/parser.c:10027
    #6 0x7f6705dfea47 in xmlParseContent__internal_alias /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/pars
    #7 0x7f6705e16d8f in xmlParseElement__internal_alias /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/pars
    #8 0x7f6705e16d8f in xmlParseDocument__internal_alias /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/par
    #9 0x7f6705e6a5ab in xmlDoRead /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/parser.c:15101
    #10 0x7f6705e6a5ab in xmlCtxtReadFile__internal_alias /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/par
    #11 0x5608cd1daee8 in parseAndPrintFile /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/xmllint.c:2332
    #12 0x5608cd1c0741 in main /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/xmllint.c:3732
    #13 0x7f6705fadb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #14 0x5608cd1ca1b9 in _start (/mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/.libs/xmllint+0x1f1b9)

0x5608cd40a230 is located 48 bytes to the left of global variable 'end' defined in 'xmllint.c:420:30' (0x5608cd40a260) of size 16
0x5608cd40a230 is located 0 bytes to the right of global variable 'buffer' defined in 'xmllint.c:525:13' (0x5608cd3fdee0) of size 50
SUMMARY: AddressSanitizer: global-buffer-overflow /mnt/hda2/suhwan/add_project/final/FINAL_TEST_ZONE/program/xxx/libxml2/entities.c:
Shadow bytes around the buggy address:
  0x0ac199a793f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ac199a79400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ac199a79410: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ac199a79420: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ac199a79430: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0ac199a79440: 00 00 00 00 00 00[f9]f9 f9 f9 f9 00 00 f9 f9
  0x0ac199a79450: f9 f9 f9 f9 00 00 f9 f9 f9 f9 f9 00 00 f9 f9
  0x0ac199a79460: f9 f9 f9 f9 04 f9 f9 f9 f9 f9 04 f9 f9 f9
  0x0ac199a79470: f9 f9 f9 f9 00 00 00 00 00 00 00 00 00 00 00
  0x0ac199a79480: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0ac199a79490: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==19600==ABORTING
```

◀   ▶

This is found by Agency for Defense Development (ADD).

⬆ Drag your designs here or click to upload.

Tasks 🎯 0	
No tasks are currently assigned. Use tasks to break down this issue into smaller parts.	

Linked items ⓘ 🗋 1	
⊖ heap-buffer-overflow in entities.c:621 #235	

Activity

⊖ **Nick Wellnhofer** closed via commit 50f06b3e 2 years ago

👤 **David Kilzer** @ddkilzer · 2 years ago   `Developer`

I used git-bisect to find that this is a regression from commit 0b19f236 by Joel Hockey joel.hockey@gmail.com. It likely requires that libxml2 be built `--without iconv --with icu` instead of the reverse.

```
commit 0b19f236a263a7b0acacd4ea84dc7237303ee3d9
Author: Joel Hockey <joel.hockey@gmail.com>
Date:   Wed Oct 25 18:11:12 2017 -0700

    Fixed ICU to set flush correctly and provide pivot buffer.

    By always setting flush=TRUE when doing multiple reads, ICU
    will not correctly handle truncated utf8 chars across read
    boundaries.

    The fix is to set flush=TRUE only on final read, and to
    provide a pivot buffer which is maintained by libxml
    between calls to ucnv_convertEx.

:100644 100644 cd019c51919f317e64fa4635abef915709e388d6 5dac8a14c5268307a20ab56d1977ebddaaec3a03 M    encoding.c
:040000 040000 87c79baa619254d783f3396aff364dbd11cc692c ba6bab3ec0519063d0491d7331402c213bed7d40 M    include
```

I think the fix to xmllint.c will only fix the issue for xmllint, not for all clients of the libxml2 library.

Edited by David Kilzer 2 years ago

👤 **Joel Hockey** @joel.hockey · 2 years ago

📎 0001-Validate-UTF8-in-xmlEncodeEntities.patch

Here is a patch to validate UTF-8 within xmlEncodeEntities and avoid buffer overflow. This should fix the issue for all clients of libxml2. I would be inclined to revert 50f06b3e so that further fuzzing in xmllint will catch any other errors inside libxml2.

My change in 0b19f236 doesn't directly relate to this code, but I suspect that by fixing the handling of truncated UTF-8 in one place, it reveals errors in other places.

Edited by Joel Hockey 2 years ago

👤 **Nick Wellnhofer** @nwellnhof · 2 years ago   `Maintainer`

I think this issue is really restricted to xmllint.

> My change in 0b19f236 doesn't directly relate to this code, but I suspect that by fixing the handling of truncated UTF-8 in one place, it reveals errors in other places.

The commit probably just changed some boundaries of multi-byte characters. If the buffer in xmllint happens to end with a complete, untruncated UTF-8 sequence, the bug isn't triggered.

In general, libxml2 validates UTF-8 when parsing and then assumes that all UTF-8 is valid. But it's always a good idea to at least avoid undefined behavior with invalid UTF-8, so thanks for the patch.

Nick Wellnhofer @nwellnhof · 1 year ago   Maintainer

For the record, Joel's patch is now applied: bf227135

Nick Wellnhofer marked this issue as related to #235 (closed) 1 year ago.

Joel Hockey mentioned in commit bf227135 1 year ago

Please register or sign in to reply