New issue

Jump to bottom

A stack-buffer-overflow occurs while parsing a file #30

Closed **tank0123** opened this issue on Jul 9, 2021 · 3 comments

tank0123 commented on Jul 9, 2021

System Configuration

- AtomicParsley version: atomicparsley-20210124.204813.840499f
- Used arguments: -T 1 -t +
- Environment (Operating system, version and so on): Ubuntu 20.04.2 64bit
- Additional information: compilation with asan

```
==34075==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fffffffd800 at pc 0x7ffff75e858d bp 0x7fffffffd540 sp 0x7fffffffcce8
WRITE of size 3936 at 0x7fffffffd800 thread T0
```

Program received signal SIGSEGV, Segmentation fault.

```
#0 0x00007ffff6ffcc50 in ?? () from /lib/x86_64-linux-gnu/libgcc_s.so.1
#1 0x00007ffff6ffe77b in _Unwind_Backtrace () from /lib/x86_64-linux-gnu/libgcc_s.so.1
#2 0x00007ffff76b4a28 in ?? () from /lib/x86_64-linux-gnu/libasan.so.5
#3 0x00007ffff75af7f7 in ?? () from /lib/x86_64-linux-gnu/libasan.so.5
#4 0x00007ffff76949ed in ?? () from /lib/x86_64-linux-gnu/libasan.so.5
#5 0x00007ffff7694363 in ?? () from /lib/x86_64-linux-gnu/libasan.so.5
#6 0x00007ffff75e85af in ?? () from /lib/x86_64-linux-gnu/libasan.so.5
#7 0x000055555555fd597 in fread (__stream=0x615000000580, __n=0x203c, __size=0x1, __ptr=0x7fffffffd6cd)
at /usr/include/x86_64-linux-gnu/bits/stdio2.h:297
#8 APar_readX (buffer=0x7fffffffd6cd "", ISObasemediafile=ISObasemediafile@entry=0x615000000580,
pos=, length=0x203c) at /home/ubuntu/tmp/atomicparsley-20210124.204813.840499f/src/util.cpp:330
#9 0x00005555555a02d0 in APar_ExtractTrackDetails (uint32_buffer=uint32_buffer@entry=0x602000000050 "",
isofile=isofile@entry=0x615000000580, track=track@entry=0x7fffffffd6a0,
track_info=track_info@entry=0x7fffffffd6b0)
at /home/ubuntu/tmp/atomicparsley-20210124.204813.840499f/src/extracts.cpp:1286
#10 0x00005555555a243b in APar_ExtractDetails (isofile=, optional_output=)
at /home/ubuntu/tmp/atomicparsley-20210124.204813.840499f/src/extracts.cpp:1638
```

I've attached the file. Please download and check the file.
2021-05-04-09_19_50_0x5b55f77d_0xb1c1261c.zip

github-actions bot commented on Jul 9, 2021

Thanks for filing an issue! Please note that this project is only passively maintained, so your best bet for getting an issue resolved is through a pull request that is easy to verify! Please read this for more information.

wez added a commit that referenced this issue on Jul 9, 2021

`avoid buffer overrun when populating track_hdlr_name` ··· ✗ `abbce70`

wez commented on Jul 9, 2021 Owner

I've pushed a speculative fix for this; looks like a missing bounds check. Please let me know how this goes!

tank0123 commented on Jul 11, 2021 Author

I checked the code you wrote.
However, there was still a memory leak, so I wrote a patch code.

Please check #31 (comment).

wez added a commit that referenced this issue on Jul 11, 2021

`fix memory leak and avoid some heap allocations` ··· ✓ `020176f`

wez closed this as completed on Jul 13, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants