

[New issue](#)[Jump to bottom](#)

Out-of-bounds Read in VC1SequenceHeader::decode_sequence_header_adv of vc1Parser.cpp #426

🔒 Closed cemonatk opened this issue on May 24, 2021 · 2 comments

Labels

bug

cemonatk commented on May 24, 2021

Greetings,

tsMuxer has an Out-of-bounds Read issue whenever runs with the PoC sample.

Found by **Cem Onat Karagun of Diesec**

System info:

```
Ubuntu 21.04
tsMuxeR version git-f6ab2a2
```

To run PoC after unzip:

```
$ ./tsmuxer global_oob
```

[global_oob.zip](#)

Chronological Function-Call Trace:

1. detectStreamReader(char const*, MPLSParser*, bool) /src/build/./tsMuxer/main.cpp:120:34
2. METADemuxer::DetectStreamReader(BufferedReaderManager&, std::__cxx11::basic_string<char, std::char_traits, std::allocator > const&, bool) /src/build/./tsMuxer/metaDemuxer.cpp:684:35
3. METADemuxer::detectTrackReader(unsigned char*, int, AbstractStreamReader::ContainerType, int, int) /src/build/./tsMuxer/metaDemuxer.cpp:765:21
4. VC1StreamReader::checkStream(unsigned char*, int) /src/build/./tsMuxer/vc1StreamReader.cpp:146:28
5. VC1SequenceHeader::decode_sequence_header() /src/build/./tsMuxer/vc1Parser.cpp:156:20
6. VC1SequenceHeader::decode_sequence_header_adv() /src/build/./tsMuxer/vc1Parser.cpp:279:37

Root Cause of The Issue:

Constant integer arrays are defined in vc1Parser.h:

```
line 39, 40:
const int ff_vc1_fps_nr[5] = {24, 25, 30, 50, 60};
const int ff_vc1_fps_dr[2] = {1000, 1001};
```

However, the array index nr is set to 7 therefore [nr-1] is larger than boundary of array ff_vc1_fps_nr.

```
(gdb) b *0x0000000000934072
Breakpoint 7 at 0x934072: file ../tsMuxer/vc1Parser.cpp, line 274.
```

```
(gdb) p nr
$7 = 7
```

```
(gdb) p ff_vc1_fps_nr
$8 = {24, 25, 30, 50, 60}
```

```
(gdb) p ff_vc1_fps_dr
$17 = {1000, 1001}
```

```
line 276:
if (nr && nr < 8 && dr && dr < 3)
```

```
line: 279:
time_base_den = ff_vc1_fps_nr[nr - 1] * 1000;
```

A similar "demo" issue is also shared in following page:

<https://github.com/google/sanitizers/wiki/AddressSanitizerExampleGlobalOutOfBounds>

I'm sharing the link above, because ASAN declares this issue as "global-buffer-overflow" but as shared in References and root-cause sections this is actually a OOB read issue.

Recommendation:

Editing size-check of "array index" within "if condition" in line 276 might fix this "particular" issue.

An additional check of index variables (dr and nr) for ">= 0" is **recommended**.

PoC Fix:

```
- if (nr && nr < 8 && dr && dr < 3)
+ if (nr && nr < 5 && nr >= 0 && dr && dr < 3 && dr >= 0)
```

References:

<https://cwe.mitre.org/data/definitions/125.html>

Address Sanitizer Output:

```
tsMuxeR version git-f6ab2a2. github.com/justdan96/tsMuxeR
=====
==1861865==ERROR: AddressSanitizer: global-buffer-overflow on address 0x00000299234 at pc 0x00000093540f bp 0x7fffd9f6e10 sp 0x7fffd9f6e08
READ of size 4 at 0x00000299234 thread T0
#0 0x93540e in VC1SequenceHeader::decode_sequence_header_adv() /src/build/./tsMuxeR/vc1Parser.cpp:279:37
#1 0x93264d in VC1SequenceHeader::decode_sequence_header() /src/build/./tsMuxeR/vc1Parser.cpp:156:20
#2 0x940715 in VC1StreamReader::checkStream(unsigned char*, int) /src/build/./tsMuxeR/vc1StreamReader.cpp:146:28
#3 0x6d090e in METADemuxer::detectTrackReader(unsigned char*, int, AbstractStreamReader::ContainerType, int, int) /src/build/./tsMuxeR/metaDemuxer.cpp:765:21
#4 0x6c7255 in METADemuxer::DetectStreamReader(BufferedReaderManager&, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, bool)
/src/build/./tsMuxeR/metaDemuxer.cpp:684:35
#5 0x5df87e in detectStreamReader(char const*, MPLSParser*, bool) /src/build/./tsMuxeR/main.cpp:120:34
#6 0x5efd05 in main /src/build/./tsMuxeR/main.cpp:698:17
#7 0x7f5f89b14564 in __libc_start_main csu/../csu/libc-start.c:332:16
#8 0x2ebded in _start (/home/X/tsMuxeR/bin/tsMuxeR+0x2ebded)

0x00000299234 is located 44 bytes to the left of global variable '<string literal>' defined in '../tsMuxeR/bitStream.h:131:9' (0x299260) of size 15
'<string literal>' is ascii string 'num <= INT_BIT'
0x00000299234 is located 0 bytes to the right of global variable 'ff_vc1_fps_nr' defined in '../tsMuxeR/vc1Parser.h:39:11' (0x299220) of size 20
SUMMARY: AddressSanitizer: global-buffer-overflow /src/build/./tsMuxeR/vc1Parser.cpp:279:37 in VC1SequenceHeader::decode_sequence_header_adv()
Shadow bytes around the buggy address:
 0x00000004b1f0: f9 f9 f9 f9 00 f9 f9 f9 f9 f9 00 02 f9 f9
 0x00000004b200: f9 f9 f9 f9 00 f9 f9 f9 f9 f9 00 06 f9 f9
 0x00000004b210: f9 f9 f9 f9 03 f9 f9 f9 f9 f9 00 05 f9 f9
 0x00000004b220: f9 f9 f9 f9 00 02 f9 f9 f9 f9 00 00 00 00
 0x00000004b230: 00 00 00 00 00 03 f9 f9 f9 f9 00 f9 f9
->0x00000004b240: f9 f9 f9 f9 00 00[04]f9 f9 f9 f9 00 07 f9 f9
 0x00000004b250: f9 f9 f9 f9 00 07 f9 f9 f9 f9 00 00 00 00
 0x00000004b260: 00 05 f9 f9 f9 f9 f9 f9 00 00 00 00 00 00
 0x00000004b270: 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x00000004b280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x00000004b290: 00 00 00 00 00 01 f9 f9 f9 f9 06 f9 f9
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
Shadow gap: cc
==1861865==ABORTING
```

jcdr428 commented on May 24, 2021

Collaborator

Hi @cemonatk , thank you for all these bug busting :)
The code actually comes from ffmpeg
https://ffmpeg.org/doxygen/2.7/vc1data_8c_source.html#I00087
https://ffmpeg.org/doxygen/2.7/vc1_8c_source.html#I00479

I will push the fix accordingly.

 jcdr428 mentioned this issue on May 24, 2021

[Bug] Incorrect values of VC-1 framerate #429


 Merged

cemonatk commented on May 24, 2021

Author

Hi @jcdr428 , you are welcome. Oh, I see... quite interesting one.
I'm glad it's fixed atm.

 xavery closed this as completed in 378377e on Jun 9, 2021

 jcdr428 added the bug label on Jun 23

Assignees

No one assigned

Labels

bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

