

New issue

[Jump to bottom](#)

## heap-buffer-overflow tsmuxer #395

🔒 Closed NigelX opened this issue on Feb 3, 2021 · 3 comments · Fixed by #396

Labels

bug

NigelX commented on Feb 3, 2021 • edited

hello, guys.I use afl-fuzz to test tsMuxer.I found a crash.

./tsmuxer poc.wav

tsMuxer: 2.6.16  
OS:ubuntu 20.04[poc.zip](#)

Asan log

```
tsMuxeR version git-c515350. github.com/justdan96/tsMuxer
=====
==1374967==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000b951 at pc 0x55b50151fa0e bp 0x7ffc47bf1430 sp 0x7ffc47bf1420
READ of size 1 at 0x6020000b951 thread T0
#0 0x55b50151fa0d in BitStreamReader::getCurVal(unsigned int*) /home/hx_server/userData/target/tsMuxer/tsMuxer/bitStream.h:62
#1 0x55b50151fa0d in BitStreamReader::setBuffer(unsigned char*, unsigned char*) /home/hx_server/userData/target/tsMuxer/tsMuxer/bitStream.h:71
#2 0x55b50151fa0d in VCISquenceHeader::decode_entry_point() /home/hx_server/userData/target/tsMuxer/tsMuxer/vc1Parser.cpp:314
#3 0x55b501531da4 in VCISStreamReader::checkStream(unsigned char*, int) /home/hx_server/userData/target/tsMuxer/tsMuxer/vc1StreamReader.cpp:139
#4 0x55b50116272b in METADemuxer::detectTrackReader(unsigned char*, int, AbstractStreamReader::ContainerType, int, int)
/home/hx_server/userData/target/tsMuxer/tsMuxer/metaDemuxer.cpp:765
#5 0x55b501170383 in METADemuxer::DetectStreamReader(BufferedReaderManager&, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, bool)
/home/hx_server/userData/target/tsMuxer/tsMuxer/metaDemuxer.cpp:684
#6 0x55b50106f372 in detectStreamReader(char const*, MPLSParser*, bool) /home/hx_server/userData/target/tsMuxer/tsMuxer/main.cpp:120
#7 0x55b500e09a13 in main /home/hx_server/userData/target/tsMuxer/tsMuxer/main.cpp:698
#8 0x7f05816a20b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x270b2)
#9 0x55b500e533ed in _start (/home/hx_server/userData/target/tsMuxer/af1_build/tsMuxer/tsmuxer+0x1943ed)

0x6020000b951 is located 0 bytes to the right of 1-byte region [0x6020000b950,0x6020000b951)
allocated by thread T0 here:
#0 0x7f0581dc7b47 in operator new[](unsigned long) (/lib/x86_64-linux-gnu/libasan.so.5+0x10fb47)
#1 0x55b5015315f2 in VC1Unit::vc1_unescape_buffer(unsigned char*, int) /home/hx_server/userData/target/tsMuxer/tsMuxer/vc1Parser.h:94
#2 0x55b5015315f2 in VC1StreamReader::checkStream(unsigned char*, int) /home/hx_server/userData/target/tsMuxer/tsMuxer/vc1StreamReader.cpp:138

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/hx_server/userData/target/tsMuxer/tsMuxer/bitStream.h:62 in BitStreamReader::getCurVal(unsigned int*)
Shadow bytes around the buggy address:
 0x0c047fff96d0: fa fa fd fd fa fa fd fa fa fa fd fd
 0x0c047fff96e0: fa fa fd fa fa fa fd fa fa fd fd fa
 0x0c047fff96f0: fa fa fd fa fa fa fd fd fa fa fd fd
 0x0c047fff9700: fa fa fd fa fa fa fd fa fa fd fd fa
 0x0c047fff9710: fa fa fd fa fa fa fd fd fa fa fd fa
->0x0c047fff9720: fa fa fd fd fa fa 00 00 fa fa[01]fa fa fa fa
 0x0c047fff9730: fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c047fff9740: fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c047fff9750: fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c047fff9760: fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c047fff9770: fa fa fa fa fa fa fa fa fa fa fa fa
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
==1374967==ABORTING
```

gdb

```
[ Legend: Modified register | Code | Heap | Stack | String ]
----- registers -----
$rax : 0x0
$rbx : 0x00007ffff793c080 -> 0x00007ffff793c080 -> [loop detected]
$rcx : 0x00007ffff79bb18b -> <raise+203> mov rax, QWORD PTR [rsp+0x108]
$rdx : 0x0
$rsp : 0x00007ffff7f7090 -> 0x0000000000000000
$rbp : 0x00007ffff7f73e0 -> 0x00007ffff7f7490 -> 0x00007ffff7f74b0 -> 0x00007ffff7f74d0 -> 0x00007ffff7f74f0 -> 0x00007ffff7f7510 -> 0x00007ffff7f7530 -> 0x00007ffff7f7550
0x00007ffff7f7570
$rsi : 0x00007ffff7f7090 -> 0x0000000000000000
$rdi : 0x2
$rip : 0x00007ffff79bb18b -> <raise+203> mov rax, QWORD PTR [rsp+0x108]
$r8 : 0x0
```

```
$r9 : 0x0007ffffff7090 → 0x0000000000000000
$r10 : 0x8
$r11 : 0x246
$r12 : 0x0007ffffff7300 → 0x0007ffff39382e1 → 0xac440000ac440001
$r13 : 0x10
$r14 : 0x0007ffff7fb000 → 0x6565726600001000
$r15 : 0x1
$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 -----
0x0007ffffff7090|+0x0000: 0x0000000000000000 ← $rsp, $rsi, $r9
0x0007ffffff7098|+0x0008: 0x0000ffff00001f80
0x0007ffffff70a0|+0x0010: 0x0000000000000000
0x0007ffffff70a8|+0x0018: 0x0007ffff7b79f40 → <_Unwind_RaiseException+0> endbr64
0x0007ffffff70b0|+0x0020: 0x4000000000000000
0x0007ffffff70b8|+0x0028: 0x0000000000000000
0x0007ffffff70c0|+0x0030: 0x0000000000000000
0x0007ffffff70c8|+0x0038: 0x0000000000000000
----- code:x86:64 -----
0x7ffff79bb17f <raise+191> mov edi, 0x2
0x7ffff79bb184 <raise+196> mov eax, 0xe
0x7ffff79bb189 <raise+201> syscall
→ 0x7ffff79bb18b <raise+203> mov rax, QWORD PTR [rsp+0x108]
0x7ffff79bb193 <raise+211> xor rax, QWORD PTR fs:0x28
0x7ffff79bb19c <raise+220> jne 0x7ffff79bb1c4 <__GI_raise+260>
0x7ffff79bb19e <raise+222> mov eax, r8d
0x7ffff79bb1a1 <raise+225> add rsp, 0x118
0x7ffff79bb1a8 <raise+232> ret
----- threads -----
[#0] Id 1, Name: "tsmuxe", stopped 0x7ffff79bb18b in __GI_raise (), reason: SIGABRT
----- trace -----
[#0] 0x7ffff79bb18b → __GI_raise(sig=0x6)
[#1] 0x7ffff799a859 → __GI_abort()
[#2] 0x7ffff7a053ee → __libc_message(action=do_abort, fmt=0x7ffff7b2f285 "%s\n")
[#3] 0x7ffff7a0d47c → malloc_printerr(str=0x7ffff7b2d4ae "free(): invalid pointer")
[#4] 0x7ffff7a0ecac → _int_free(av=<optimized out>, p=<optimized out>, have_lock=0x0)
[#5] 0x55555575961a → VC1Unit::~VC1Unit()()
[#6] 0x5555557596f6 → VC1SequenceHeader::~VC1SequenceHeader()()
[#7] 0x555555759842 → VC1StreamReader::~VC1StreamReader()()
[#8] 0x55555574fc24 → METADemuxer::detectTrackReader(unsigned char*, int, AbstractStreamReader::ContainerType, int, int)()
[#9] 0x55555574ee31 → METADemuxer::DetectStreamReader(BufferedReaderManager&, std::cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, bool)()
-----
```

HX from Topsec alpha Security Team

jcdr428 commented on Feb 3, 2021

Collaborator

@NigelX your poc.wav is actually an incomplete mp4 file, is this intentional ?

xavery commented on Feb 3, 2021 • edited

Contributor

I'd say so, given how it's actually the fuzzers' job to generate data that's almost-valid-but-not-really, and thus trigger crashes in code paths which are usually left alone. The extension is irrelevant in this case.

 xavery mentioned this issue on Feb 3, 2021

Fix an invalid delete error in vc1Parser.h #396



 xavery closed this as completed in #396 on Feb 3, 2021

 xavery added a commit that referenced this issue on Feb 3, 2021


 Fix an invalid delete error in vc1Parser.h (#396) ...

✓ 0821aa6

xavery commented on Feb 3, 2021

Contributor

Should be fixed now, thanks.

 jcdr428 added the bug label on Jun 23

Assignees

No one assigned

Labels

bug

Projects


None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

 **Fix an invalid delete error in vc1Parser.h**  
justdan96/tsMuxer

---

3 participants

