

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

heap overflow in wasm-interp #1981

✓ Closed Q1IQ opened this issue on Sep 5 · 1 comment

Q1IQ commented on Sep 5

Environment

```
OS      : Linux ubuntu 5.15.0-46-generic #49~20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
x86_64 x86_64 GNU/Linux
Commit  : 3054d61f703d609995798f872fc86b462617c294
Version : 1.0.29
Build   : make clang-debug-asan
```

Proof of concept

poc-interp-1.wasm

[poc-interp-1.wasm.zip](#)

Stack dump

```
/wabt/out/clang/Debug/asan/wasm-interp --enable-all ./poc-interp-1.wasm
=====
==3163982==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6060000e318 at pc
0x000000509b36 bp 0x7ffeb0802e90 sp 0x7ffeb0802e88
READ of size 8 at 0x6060000e318 thread T0
#0 0x509b35 in std::vector<wabt::Type, std::allocator<wabt::Type>>::size() const
/usr/lib/gcc/x86_64-linux-gnu/10/../../../../include/c++/10/bits/stl_vector.h:919:40
#1 0x576a36 in wabt::interp::(anonymous
namespace)::BinaryReaderInterp::GetReturnCallDropKeepCount(wabt::interp::FuncType const&, unsigned
int, unsigned int*, unsigned int*) /wabt/out/clang/Debug/asan/../../../../src/interp/binary-
reader-interp.cc:445:58
#2 0x56955c in wabt::interp::(anonymous
namespace)::BinaryReaderInterp::OnReturnCallIndirectExpr(unsigned int, unsigned int)
/wabt/out/clang/Debug/asan/../../../../src/interp/binary-reader-interp.cc:1176:3
#3 0x6ead11 in wabt::(anonymous namespace)::BinaryReader::ReadInstructions(bool, unsigned
long, wabt::Opcode*) /wabt/out/clang/Debug/asan/../../../../src/binary-reader.cc:937:9
```

```

#4 0x6ff84e in wabt::(anonymous namespace)::BinaryReader::ReadFunctionBody(unsigned long)
/wabt/out/clang/Debug/asan/../../../../src/binary-reader.cc:667:3
#5 0x6c2f98 in wabt::(anonymous namespace)::BinaryReader::ReadCodeSection(unsigned long)
/wabt/out/clang/Debug/asan/../../../../src/binary-reader.cc:2766:7
#6 0x6b0861 in wabt::(anonymous namespace)::BinaryReader::ReadSections(wabt::(anonymous
namespace)::BinaryReader::ReadSectionsOptions const&)
/wabt/out/clang/Debug/asan/../../../../src/binary-reader.cc:2920:26
#7 0x6ada2f in wabt::(anonymous namespace)::BinaryReader::ReadModule(wabt::(anonymous
namespace)::BinaryReader::ReadModuleOptions const&)
/wabt/out/clang/Debug/asan/../../../../src/binary-reader.cc:2981:3
#8 0x6aca08 in wabt::ReadBinary(void const*, unsigned long, wabt::BinaryReaderDelegate*,
wabt::ReadBinaryOptions const&) /wabt/out/clang/Debug/asan/../../../../src/binary-
reader.cc:2998:17
#9 0x54f132 in wabt::interp::ReadBinaryInterp(std::basic_string_view<char,
std::char_traits<char>>, void const*, unsigned long, wabt::ReadBinaryOptions const&,
std::vector<wabt::Error, std::allocator<wabt::Error>>*, wabt::interp::ModuleDesc*)
/wabt/out/clang/Debug/asan/../../../../src/interp/binary-reader-interp.cc:1603:10
#10 0x4f6aed in ReadModule(char const*, std::vector<wabt::Error,
std::allocator<wabt::Error>>*, wabt::interp::RefPtr<wabt::interp::Module>*)
/wabt/out/clang/Debug/asan/../../../../src/tools/wasm-interp.cc:207:3
#11 0x4f100a in ReadAndRunModule(char const*)
/wabt/out/clang/Debug/asan/../../../../src/tools/wasm-interp.cc:234:19
#12 0x4f0427 in ProgramMain(int, char**)
/wabt/out/clang/Debug/asan/../../../../src/tools/wasm-interp.cc:329:25
#13 0x4f14a1 in main /wabt/out/clang/Debug/asan/../../../../src/tools/wasm-interp.cc:335:10
#14 0x7ff3cc7e6082 in __libc_start_main /build/glibc-SzIz7B/glibc-2.31/csu/./csu/libc-
start.c:308:16
#15 0x43e39d in _start (/wabt/out/clang/Debug/asan/wasm-interp+0x43e39d)

```

Address 0x60600000e318 is a wild pointer inside of access range of size 0x000000000008.

SUMMARY: AddressSanitizer: heap-buffer-overflow /usr/lib/gcc/x86_64-linux-
gnu/10/../../../../include/c++/10/bits/stl_vector.h:919:40 in std::vector<wabt::Type,
std::allocator<wabt::Type>>::size() const

Shadow bytes around the buggy address:

```

0x0c0c7fff9c10: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9c20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9c30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9c40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9c50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c0c7fff9c60: fa fa fa[fa]fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9c70: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9c80: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9c90: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9ca0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c0c7fff9cb0: fa fa fa fa 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
==3163982==ABORTING
```

keithw commented on Sep 17

Collaborator

Thank you for finding and reporting this. It appears to be fixed by [#1931](#).

I'm a little curious about your fuzzing approach -- have you written a better fuzz target than is available at <https://github.com/google/oss-fuzz/tree/master/projects/wabt> ? If so, would you be interested to contribute it upstream? It seems like you are doing a better job of fuzzing wabt than the existing oss-fuzz targets are.



keithw closed this as completed on Sep 17



This was referenced on Sep 17

Out-of-bound read in OnReturnCallExpr->GetReturnCallDropKeepCount #1982

✓ Closed

Out-of-bound read in OnReturnCallIndirectExpr->GetReturnCallDropKeepCount #1983

✓ Closed

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

