heap overflow in tinyexr::DecodePixelData #167

Closed sleicasper opened this issue on Jun 14 · 8 comments

Labels enhancement

sleicasper commented on Jun 14 • edited

desc

There is a heap based buffer overflow in tinyexr::DecodePixelData before 20220506 that could cause remote code execution depending on the usage of this program.

asan output

```
==2363537==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x629000009210 at pc
0x000000563bd4 bp 0x7fffffffc4b0 sp 0x7fffffffc4a8
READ of size 1 at 0x629000009210 thread T0
    #0 0x563bd3 in tinyexr::cpy4(float*, float const*) /tinyexr/./BUILD/tinyexr.h:759:12
    #1 0x563bd3 in tinyexr::DecodePixelData(unsigned char**, int const*, unsigned char const*,
unsigned long, int, int, int, int, int, int, int, int, int, unsigned long, unsigned long, TEXRAttribute
const*, unsigned long, TEXRChannelInfo const*, std::vector<unsigned long, std::allocator<unsigned
long> > const&) /tinyexr/./BUILD/tinyexr.h:3593:13
    #2 0x505f79 in tinyexr::DecodeTiledPixelData(unsigned char**, int*, int*, int const*, unsigned
char const*, unsigned long, int, int, int, int, int, int, int, int, int, unsigned long, unsigned long,
TEXRAttribute const*, unsigned long, TEXRChannelInfo const*, std::vector<unsigned long,
std::allocator<unsigned long> > const&) /tinyexr/./BUILD/tinyexr.h:4115:10
    #3 0x505f79 in tinyexr::DecodeTiledLevel(TEXRImage*, TEXRHeader const*, tinyexr::OffsetData
const&, std::vector<unsigned long, std::allocator<unsigned long> > const&, int, unsigned char
const*, unsigned long, std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >*) /tinyexr/./BUILD/tinyexr.h:4841:16
    #4 0x504abc in tinyexr::DecodeChunk(TEXRImage*, TEXRHeader const*, tinyexr::OffsetData const&,
unsigned char const*, unsigned long, std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >*) /tinyexr/./BUILD/tinyexr.h:5015:19
    #5 0x519246 in tinyexr::DecodeEXRImage(TEXRImage*, TEXRHeader const*, unsigned char const*,
unsigned char const*, unsigned long, char const**) /tinyexr/./BUILD/tinyexr.h:5756:15
    #6 0x519246 in LoadEXRImageFromMemory /tinyexr/./BUILD/tinyexr.h:6444:10
    #7 0x53f3bf in LLVMFuzzerTestOneInput /tinyexr/./SRC/test/fuzzer/fuzz.cc:20:9
    #8 0x4fbaad in fuzzfile(char*) /tinyexr/../../../harness/aflharness.cc:35:5
```

```
    #9 0x4fbc06 in main /tinyexr/../../../harness/aflharness.cc:52:13
    #10 0x7ffff7a51082 in __libc_start_main /build/glibc-SzIz7B/glibc-2.31/csu/../csu/libc-
start.c:308:16
    #11 0x41e65d in _start (/tinyexr/fuzzrun/harness+0x41e65d)

0x629000009210 is located 0 bytes to the right of 16400-byte region
[0x629000005200,0x629000009210)
allocated by thread T0 here:
    #0 0x4f8c17 in operator new(unsigned long) /fuzz/fuzzdeps/llvm-project-11.0.0/compiler-
rt/lib/asan/asan_new_delete.cpp:99:3
    #1 0x55b2b2 in __gnu_cxx::new_allocator<unsigned char>::allocate(unsigned long, void const*)
/usr/lib/gcc/x86_64-linux-gnu/9/../../../../include/c++/9/ext/new_allocator.h:114:27
    #2 0x55b2b2 in std::allocator_traits<std::allocator<unsigned char>
>::allocate(std::allocator<unsigned char>&, unsigned long) /usr/lib/gcc/x86_64-linux-
gnu/9/../../../../include/c++/9/bits/alloc_traits.h:443:20
    #3 0x55b2b2 in std::_Vector_base<unsigned char, std::allocator<unsigned char>
>::_M_allocate(unsigned long) /usr/lib/gcc/x86_64-linux-
gnu/9/../../../../include/c++/9/bits/stl_vector.h:343:20
    #4 0x55b2b2 in std::_Vector_base<unsigned char, std::allocator<unsigned char>
>::_M_create_storage(unsigned long) /usr/lib/gcc/x86_64-linux-
gnu/9/../../../../include/c++/9/bits/stl_vector.h:358:33
    #5 0x55b2b2 in std::_Vector_base<unsigned char, std::allocator<unsigned char>
>::_Vector_base(unsigned long, std::allocator<unsigned char> const&) /usr/lib/gcc/x86_64-linux-
gnu/9/../../../../include/c++/9/bits/stl_vector.h:302:9
    #6 0x55b2b2 in std::vector<unsigned char, std::allocator<unsigned char> >::vector(unsigned
long, std::allocator<unsigned char> const&) /usr/lib/gcc/x86_64-linux-
gnu/9/../../../../include/c++/9/bits/stl_vector.h:508:9
    #7 0x55b2b2 in tinyexr::DecodePixelData(unsigned char**, int const*, unsigned char const*,
unsigned long, int, int, int, int, int, int, int, int, int, unsigned long, unsigned long, TEXRAttribute
const*, unsigned long, TEXRChannelInfo const*, std::vector<unsigned long, std::allocator<unsigned
long> > const&) /tinyexr/./BUILD/tinyexr.h:3484:32
    #8 0x505f79 in tinyexr::DecodeTiledPixelData(unsigned char**, int*, int*, int const*, unsigned
char const*, unsigned long, int, int, int, int, int, int, int, int, unsigned long, unsigned long,
TEXRAttribute const*, unsigned long, TEXRChannelInfo const*, std::vector<unsigned long,
std::allocator<unsigned long> > const&) /tinyexr/./BUILD/tinyexr.h:4115:10
    #9 0x505f79 in tinyexr::DecodeTiledLevel(TEXRImage*, TEXRHeader const*, tinyexr::OffsetData
const&, std::vector<unsigned long, std::allocator<unsigned long> > const&, int, unsigned char
const*, unsigned long, std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >*) /tinyexr/./BUILD/tinyexr.h:4841:16

SUMMARY: AddressSanitizer: heap-buffer-overflow /tinyexr/./BUILD/tinyexr.h:759:12 in
tinyexr::cpy4(float*, float const*)
Shadow bytes around the buggy address:
  0x0c527fff91f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c527fff9200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c527fff9210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c527fff9220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c527fff9230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c527fff9240: 00 00[fa]fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff9250: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff9260: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff9270: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff9280: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c527fff9290: 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
  Shadow gap:              cc
==2363537==ABORTING
```

reproduce

- compile this project using address sanitizer
- run `./test/fuzzer ./poc`

 **syoyo** added the **invalid** label on Jun 15

syoyo commented on Jun 15 — Owner

No issue observed in `master` `0c48a75`

```
$ clang++ -fsanitize=address -Wno-padded -Weverything -Werror -Wall -Wextra -std=c++11 -g -O0 -
DTINYEXR_USE_MINIZ=1 -DTINYEXR_USE_PIZ=1 -I./deps/miniz -o test_tinyexr test_tinyexr.cc miniz.o
$ ./test_tinyexr poc
Header err. code -1
```

 **syoyo** closed this as completed on Jun 15

sleicasper commented on Jun 15 — Author

Well, I can still reproduce this issue

syoyo commented on Jun 15 — Owner

You need to post compilation procedure in detail.

sleicasper commented on Jun 15 — Author

> You need to post compilation procedure in detail.

```
clang -c deps/miniz/miniz.c -o miniz.o
clang++ -fsanitize=address -Wno-padded -Weverything -Werror -Wall -Wextra -std=c++11 -g -O0 -
DTINYEXR_USE_MINIZ=1 -DTINYEXR_USE_PIZ=1 -I./deps/miniz -o test_tinyexr test_tinyexr.cc miniz.o
./test_tinyexr poc
```

syoyo commented on Jun 15 — Owner

```
clang++ -fsanitize=address -Wno-padded -Weverything -Werror -Wall -Wextra -std=c++11 -g -O0 -
DTINYEXR_USE_MINIZ=1 -DTINYEXR_USE_PIZ=1 -I./deps/miniz -o test_tinyexr test_tinyexr.cc miniz.o
./test_tinyexr poc
```

Still no issue with it. Reports `Header err. code -1`

Found you are attaching wrong POC file. Seems a Core audio file? Magic header starts with `caff`

```
caff^@^A^@^@desc^@^@^@^@ô^@^@ @å<88><80>^@^@^@^@lpcm^@^@^@^A^@^@^@^
^Aencoder^@Lcvf58.35.101^@data^@^@^@^@^@^@^@^P^@^B^@^@^@^@ÿcaff^@^
^@^@^@^@^L^@d^@^A^@^@^@^@^@^@^@^@info^@^@^@^@^@^@^@B^@^@^@^FTLEN^@0
@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@æÿ^@^@^
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^
@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^
A^@ÿÿ^@^@^A^@^A^@^@^@^@^@^@^@^@^@^A^@^B^@^A^@ÿÿ^A^@^B^@^A^@^@^@^@^@
ÿýÿÿÿùÿúÿ^H^@^B^@øÿ^E^@ ^@ÿÿ^A^@        ^@^H^@ýÿûÿ
^@^G^@÷ÿýÿ^C^@þÿ^A^@^C^@ýÿþÿ^F^@^K^@^C^@ûÿ^H^@^M^@^A^@^C^@      ^@^A
@^S^@àÿëÿ(^@
^@ßÿûÿ^W^@^N^@ìÿõÿ#^@^C^@ëÿ^^^@^Y^@^E^@^N^@^G^@-^@)^@íÿ^[^@4^@^C^@^C
```

sleicasper commented on Jun 16 — Author

> ```
> clang++ -fsanitize=address -Wno-padded -Weverything -Werror -Wall -Wextra -std=c++11 -g -O0 -
> DTINYEXR_USE_MINIZ=1 -DTINYEXR_USE_PIZ=1 -I./deps/miniz -o test_tinyexr test_tinyexr.cc
> miniz.o
> ./test_tinyexr poc
> ```
>
> Still no issue with it. Reports `Header err. code -1`
>
> Found you are attaching wrong POC file. Seems a Core audio file? Magic header starts with `caff`
>
> ```
> caff^@^A^@^@desc^@^@^@^@ô^@^@ @å<88><80>^@^@^@^@lpcm^@^@^@^A^@^@^@^H
> ^Aencoder^@Lcvf58.35.101^@data^@^@^@^@^@^@^@^P^@^B^@^@^@^@^@ÿcaff^@
> ^@^@^@^@^L^@d^@^A^@^@^@^@^@^@^@^@info^@^@^@^@^@^@^@B^@^@^@^FTLEN^@0
> @^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@æÿ^@^@
> ^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
> @^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
> ^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^A^@^@^@^@^@^@^@^@^@^@^@^@^A^@^@^
> A^@ÿÿ^@^@^A^@^A^@^@^@^@^@^@^@^@^@^A^@^B^@^A^@ÿÿ^A^@^B^@^A^@^@^@^@^@
> ÿýÿÿÿùÿúÿ^H^@^B^@øÿ^E^@ ^@ÿÿ^A^@        ^@^H^@ýÿûÿ
> ^@^G^@÷ÿýÿ^C^@þÿ^A^@^C^@ýÿþÿ^F^@^K^@^C^@ûÿ^H^@^M^@^A^@^C^@      ^@^A
> @^S^@àÿëÿ(^@
> ^@ßÿûÿ^W^@^N^@ìÿõÿ#^@^C^@ëÿ^^^@^Y^@^E^@^N^@^G^@-^@)^@íÿ^[^@4^@^C^@^C
> ```

your are right.

new poc:

poc.zip

syoyo commented on Jun 16 • edited ▾ Owner

Thanks! Confirmed the issue is now reproducible.

Your PR to fix the issue is much appreciated.

syoyo reopened this on Jun 16

syoyo added **enhancement** and removed **invalid** labels on Jun 16

syoyo commented on Jun 28 Owner

Close the issue to avoid CVE FUD

syoyo closed this as completed on Jun 28

roehling added a commit to roehling/tinyexr that referenced this issue on Sep 8

`Fix out of bounds access in DecodePixelData` ··· `9c602a3`

roehling mentioned this issue on Sep 8

Fix out of bounds access in DecodePixelData #175

Merged

Assignees

No one assigned

Labels

enhancement

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants