

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

Heap-buffer-overflow still exists in the rleUncompress #169

✓ Closed 0xdd96 opened this issue on Jul 5 · 3 comments

Labels

bug

0xdd96 commented on Jul 5

Contributor

Describe the issue

Heap-buffer-overflow still exists in the `rleUncompress`.

This is similar to issue [#112](#), but it seems that the patch [58a6258](#) has not fully fixed them.

To Reproduce

Environment

- OS: Ubuntu 16.04.7 LTS
- Compiler: gcc version 5.4.0

version: latest commit [0647fb3](#)

poc: [poc](#)

Steps to reproduce the behavior:

1. Compile TinyEXR with Address Sanitizer

```
CFLAGS="-g -O0 -fsanitize=address" CXXFLAGS="-g -O0 -fsanitize=address" cmake -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Release ..
```

2. run `./test_tinyexr ./poc`

Here is the trace reported by ASAN:

```
root@d8a714203f6e:~# ./test_tinyexr poc
```

```
=====
```

```
==14886==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x619000006337 at pc  
0x00000040c22d bp 0x7fffffffcb50 sp 0x7fffffffcb40
```

```
READ of size 1 at 0x619000006337 thread T0
```

```
#0 0x40c22c in rleUncompress tinyexr/tinyexr.h:1522  
#1 0x40c22c in DecompressRle tinyexr/tinyexr.h:1625  
#2 0x40c22c in DecodePixelData tinyexr/tinyexr.h:3786  
#3 0x411318 in DecodeChunk tinyexr/tinyexr.h:5176  
#4 0x41a3e9 in DecodeEXRImage tinyexr/tinyexr.h:5776  
#5 0x41dcc7 in LoadEXRImageFromMemory tinyexr/tinyexr.h:6465  
#6 0x41dcc7 in LoadEXRImageFromFile tinyexr/tinyexr.h:6442  
#7 0x4288ae in LoadEXRWithLayer tinyexr/tinyexr.h:5954  
#8 0x40502f in LoadEXR tinyexr/tinyexr.h:5902  
#9 0x40502f in test_main tinyexr/test_tinyexr.cc:223  
#10 0x40502f in main tinyexr/test_tinyexr.cc:194  
#11 0x7ffff652883f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2083f)  
#12 0x4053b8 in _start (tinyexr/build-gcc/test_tinyexr+0x4053b8)
```

```
0x619000006337 is located 0 bytes to the right of 951-byte region [0x619000005f80,0x619000006337)  
allocated by thread T0 here:
```

```
#0 0x7ffff6f03532 in operator new(unsigned long) (/usr/lib/x86_64-linux-  
gnu/libasan.so.2+0x99532)  
#1 0x41dc55 in __gnu_cxx::new_allocator<unsigned char>::allocate(unsigned long, void const*)  
/usr/include/c++/5/ext/new_allocator.h:104  
#2 0x41dc55 in std::allocator_traits<std::allocator<unsigned char>  
>::allocate(std::allocator<unsigned char>&, unsigned long)  
/usr/include/c++/5/bits/alloc_traits.h:491  
#3 0x41dc55 in std::_Vector_base<unsigned char, std::allocator<unsigned char>  
>::_M_allocate(unsigned long) /usr/include/c++/5/bits/stl_vector.h:170  
#4 0x41dc55 in std::_Vector_base<unsigned char, std::allocator<unsigned char>  
>::_M_create_storage(unsigned long) /usr/include/c++/5/bits/stl_vector.h:185  
#5 0x41dc55 in std::_Vector_base<unsigned char, std::allocator<unsigned char>  
>::_Vector_base(unsigned long, std::allocator<unsigned char> const&)  
/usr/include/c++/5/bits/stl_vector.h:136  
#6 0x41dc55 in std::vector<unsigned char, std::allocator<unsigned char> >::vector(unsigned  
long, std::allocator<unsigned char> const&) /usr/include/c++/5/bits/stl_vector.h:278  
#7 0x41dc55 in LoadEXRImageFromFile tinyexr/tinyexr.h:6432
```

```
SUMMARY: AddressSanitizer: heap-buffer-overflow tinyexr/tinyexr.h:1522 rleUncompress  
Shadow bytes around the buggy address:
```

```
0x0c327fff8c10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0c327fff8c20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0c327fff8c30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0c327fff8c40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0c327fff8c50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
=>0x0c327fff8c60: 00 00 00 00 00 00 00[07]fa fa fa fa fa fa fa fa fa  
0x0c327fff8c70: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa  
0x0c327fff8c80: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa  
0x0c327fff8c90: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa  
0x0c327fff8ca0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa  
0x0c327fff8cb0: 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
Heap right redzone:     fb
Freed heap region:      fd
Stack left redzone:     f1
Stack mid redzone:      f2
Stack right redzone:    f3
Stack partial redzone:  f4
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
==14886==ABORTING
```

 **0xdd96** added a commit to 0xdd96/tinyexr that referenced this issue on Jul 6



Add bounds check to address [syoyo#169](#)

cc1b199

0xdd96 commented on Jul 6

Contributor

Author

Before calling `rleUncompress`, `src_size=0x64` in the PoC, bypassing the check (line 1616) introduced in the patch for [#112](#).

[tinyexr/tinyexr.h](#)

Lines 1614 to 1618 in 0647fb3

```
1614     // Workaround for issue #112.
1615     // TODO(syoyo): Add more robust out-of-bounds check in `rleUncompress`.
1616     if (src_size <= 2) {
1617         return false;
1618     }
```

However, `inLength` is still set to -1 in line 1518 and heap buffer overflow in line 1522.

[tinyexr/tinyexr.h](#)

Lines 1501 to 1527 in 0647fb3

```
1501     static int rleUncompress(int inLength, int maxLength, const signed char in[],
1502                               char out[]) {
1503         char *outStart = out;
1504
1505         while (inLength > 0) {
1506             if (*in < 0) {
1507                 int count = -(static_cast<int>)(*in++));
1508                 inLength -= count + 1;
```

1509

1510

// Fixes #116: Add bounds check to in buffer.

1511

if ((0 > (maxLength -= count)) || (inLength < 0)) return 0;

1512



It's better to check the buffer boundary before calling `memset`, just like [#117](#).



syoyo added the `bug` label on Jul 6

syoyo commented on Jul 6

Owner

Thanks! I can reproduce the issue.

And also thank you for the PR. Will review it soon.



syoyo added a commit that referenced this issue on Jul 6



Merge pull request [#170](#) from 0xdd96/master ...

✓ 82984a3



syoyo mentioned this issue on Jul 6

Add bounds check to address #169 [#170](#)

Merged

syoyo commented on Jul 14

Owner

PR has been merged!



syoyo closed this as completed on Jul 14



syoyo mentioned this issue on Jul 14

Vulnerability found by OSS-Fuzz [syoyo/tinygltf#364](#)

Closed



syoyo mentioned this issue on Aug 19

Security concern [syoyo/tinygltf#370](#)

Assignees

No one assigned

Labels

bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

