New issue

Jump to bottom

AddressSanitizer: heap-use-after-free in ucompthread() stream.c:1538 #199

Closed **Clingto** opened this issue on May 19, 2021 · 3 comments

Clingto commented on May 19, 2021 • edited

System info:
Ubuntu 16.04.6 LTS, X64, gcc 5.4.0, lrzip (latest master 465afe8)
I think it is probably due to an imcomplete fix of #164 (imcomplete patch)

Compile Command:

```
$ chmod a+x mkinstalldirs
make distclean
./autogen.sh

mkdir -p build/bin
CC="gcc -fsanitize=address -fno-omit-frame-pointer -g" CXX="g++ -fsanitize=address -fno-omit-frame-pointer -g" ./configure --enable-static-bin --disable-shared
make -j
```

Run Command:

```
$ lrzip -t $POC
```

POC file:
https://github.com/Clingto/POC/blob/master/MSA/lrzip/lrzip-602-ucompthread-UAF

https://github.com/Clingto/POC/blob/master/MSA/lrzip/uaf-110-561

https://github.com/Clingto/POC/blob/master/MSA/lrzip/uaf-147-449

ASAN info:

```
==17630==ERROR: AddressSanitizer: heap-use-after-free on address 0x61b00001f200 at pc 0x000000420cbf bp 0x7f61990fdd60 sp 0x7f61990fdd50
READ of size 1 at 0x61b00001f200 thread T3

    #0 0x420cbe in ucompthread  test/lrzip-uaf/git/build_asan/stream.c:1538
    #1 0x7f619cddf6b9 in start_thread (/lib/x86_64-linux-gnu/libpthread.so.0+0x76b9)
    #2 0x7f619c27441c in clone (/lib/x86_64-linux-gnu/libc.so.6+0x10741c)

0x61b00001f200 is located 128 bytes inside of 1632-byte region [0x61b00001f180,0x61b00001f7e0)
freed by thread T0 here:
    #0 0x7f619d8f12ca in __interceptor_free (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x982ca)
    #1 0x41d2ca in clear_rulist  test/lrzip-uaf/git/build_asan/runzip.c:255
    #2 0x41d2ca in runzip_chunk  test/lrzip-uaf/git/build_asan/runzip.c:383
    #3 0x41d2ca in runzip_fd  test/lrzip-uaf/git/build_asan/runzip.c:403

previously allocated by thread T0 here:
    #0 0x7f619d8f179a in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x9879a)
    #1 0x425afd in open_stream_in  test/lrzip-uaf/git/build_asan/stream.c:1083

Thread T3 created by T0 here:
    #0 0x7f619d88f253 in pthread_create (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x36253)
    #1 0x420df4 in create_pthread  test/lrzip-uaf/git/build_asan/stream.c:125

SUMMARY: AddressSanitizer: heap-use-after-free  test/lrzip-uaf/git/build_asan/stream.c:1538 ucompthread
Shadow bytes around the buggy address:
  0x0c367fffbdf0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c367fffbe00: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c367fffbe10: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c367fffbe20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c367fffbe30: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
=>0x0c367fffbe40:[fd]fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe50: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe60: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe70: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe80: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe90: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
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
==17630==ABORTING
```

ckolivas commented on Feb 25 Owner

Unable to reproduce in master.

ckolivas commented on Feb 25 — Owner

This should have been addressed by tackling a similar error.

ckolivas closed this as completed on Feb 25

Clingto commented on Jul 23 — Author

> Unable to reproduce in master.
> Hi, I still reproduce the bug (in the `465afe8`) and add two more POCs.
> I don't know if it is because the multi-thread, maybe you can run it for more times (such as ten times) and test.

```
=================================================================
==23189==ERROR: AddressSanitizer: heap-use-after-free on address 0x61b00001f1e8 at pc 0x000000420c87 bp 0x7ffff26fdd60 sp 0x7ffff26fdd50
READ of size 8 at 0x61b00001f1e8 thread T3
    #0 0x420c86 in zpaq_decompress_buf test/lrzip-uaf/git/build_asan/stream.c:449
    #1 0x420c86 in ucompthread test/lrzip-uaf/git/build_asan/stream.c:1553
    #2 0x7ffff63f06b9 in start_thread (/lib/x86_64-linux-gnu/libpthread.so.0+0x76b9)
    #3 0x7ffff588551c in clone (/lib/x86_64-linux-gnu/libc.so.6+0x10751c)

0x61b00001f1e8 is located 104 bytes inside of 1632-byte region [0x61b00001f180,0x61b00001f7e0)
freed by thread T0 here:
    #0 0x7ffff6f022ca in __interceptor_free (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x982ca)
    #1 0x41d2ca in clear_rulist test/lrzip-uaf/git/build_asan/runzip.c:255
    #2 0x41d2ca in runzip_chunk test/lrzip-uaf/git/build_asan/runzip.c:383
    #3 0x41d2ca in runzip_fd test/lrzip-uaf/git/build_asan/runzip.c:403

previously allocated by thread T0 here:
    #0 0x7ffff6f0279a in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x9879a)
    #1 0x425afd in open_stream_in test/lrzip-uaf/git/build_asan/stream.c:1083

Thread T3 created by T0 here:
    #0 0x7ffff6ea0253 in pthread_create (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x36253)
    #1 0x420df4 in create_pthread test/lrzip-uaf/git/build_asan/stream.c:125

SUMMARY: AddressSanitizer: heap-use-after-free /home/aota05/yyp/new_bug/test/lrzip-uaf/git/build_asan/stream.c:449 zpaq_decompress_buf
Shadow bytes around the buggy address:
  0x0c367fffbde0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fffbdf0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fffbe00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fffbe10: 00 00 00 00 00 00 00 00 00 00 00 00 00 fa fa fa
  0x0c367fffbe20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c367fffbe30: fd fd fd fd fd fd fd fd fd fd fd fd[fd]fd fd
  0x0c367fffbe40: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe50: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe60: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe70: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe80: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
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
==23189==ABORTING



Failed to decompress buffer - lzmaerr=1
=================================================================
==16420==ERROR: AddressSanitizer: heap-use-after-free on address 0x61b00001f1e0 at pc 0x000000420d3f bp 0x7ffff26fdd60 sp 0x7ffff26fdd50
READ of size 8 at 0x61b00001f1e0 thread T3
    #0 0x420d3e in lzma_decompress_buf test/lrzip-uaf/git/build_asan/stream.c:561
    #1 0x420d3e in ucompthread test/lrzip-uaf/git/build_asan/stream.c:1541
    #2 0x7ffff63f06b9 in start_thread (/lib/x86_64-linux-gnu/libpthread.so.0+0x76b9)
    #3 0x7ffff588551c in clone (/lib/x86_64-linux-gnu/libc.so.6+0x10751c)

0x61b00001f1e0 is located 96 bytes inside of 1632-byte region [0x61b00001f180,0x61b00001f7e0)
freed by thread T0 here:
    #0 0x7ffff6f022ca in __interceptor_free (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x982ca)
    #1 0x41d2ca in clear_rulist test/lrzip-uaf/git/build_asan/runzip.c:255
    #2 0x41d2ca in runzip_chunk test/lrzip-uaf/git/build_asan/runzip.c:383
    #3 0x41d2ca in runzip_fd test/lrzip-uaf/git/build_asan/runzip.c:403

previously allocated by thread T0 here:
    #0 0x7ffff6f0279a in __interceptor_calloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x9879a)
    #1 0x425afd in open_stream_in test/lrzip-uaf/git/build_asan/stream.c:1083

Thread T3 created by T0 here:
    #0 0x7ffff6ea0253 in pthread_create (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x36253)
    #1 0x420df4 in create_pthreadtest/lrzip-uaf/git/build_asan/stream.c:125

SUMMARY: AddressSanitizer: heap-use-after-free test/lrzip-uaf/git/build_asan/stream.c:561 lzma_decompress_buf
Shadow bytes around the buggy address:
  0x0c367fffbde0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fffbdf0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fffbe00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c367fffbe10: 00 00 00 00 00 00 00 00 00 00 00 00 00 fa fa fa
  0x0c367fffbe20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c367fffbe30: fd fd fd fd fd fd fd fd fd fd fd fd[fd]fd fd fd
```

```
  0x0c367fffbe40: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe50: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe60: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe70: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c367fffbe80: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
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
==16420==ABORTING
```

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants