ckolivas / lrzip Public

Code | Issues 15 | Pull requests 2 | Actions | Projects | Wiki

New issue

Jump to bottom

Segmentation fault caused by null pointer dereference during multithread processing in ucompthread, stream.c:1523 #164

Closed 5hadowblad3 opened this issue on Sep 2, 2020 · 5 comments

5hadowblad3 commented on Sep 2, 2020 • edited

Hi, there.

There is a segmentation caused by null pointer dereference that leads to a fatal error during the execution in the newest master branch `597be1f` .
Here is a brief explanation:

```
1507 static void *ucompthread(void *data)
1508 {
1509     stream_thread_struct *s = data;
1510     rzip_control *control = s->control;
1511     int waited = 0, ret = 0, i = s->i;
1512     struct uncomp_thread *uci;
1513
1514     dealloc(data);                    obtain from thread pool
1515     uci = &ucthread[i];
1516
1517     if (unlikely(setpriority(PRIO_PROCESS, 0, control->nice_val) == -1)) {
1518         print_err("Warning, unable to set thread nice value %d...Resetting to %d\n", control->nice_val, control->current_priority);
1519         setpriority(PRIO_PROCESS, 0, (control->nice_val=control->current_priority));
1520     }
1521                                       uncheck nullity before dereference
1522 retry:
1523     if (uci->c_type != CTYPE_NONE) {
1524         switch (uci->c_type) {
1525             case CTYPE_LZMA:
1526                 ret = lzma_decompress_buf(control, uci);
1527                 break;
1528             case CTYPE_LZO:
```

This is the output during execution:

```
Decompressing...
Bad checksum: 0x5b496f91 - expected: 0x2000210c
Fatal error - exiting
Segmentation fault
```

To reproduce, run:

```
lrzip -t seg-stream1523
```

POC (unzip first):
seg-stream1523.zip

Here is the trace reported by ASAN:

```
==161258==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000080 (pc 0x00000043f8d8 bp 0x0000007cd680 sp 0x7f811dafdd80 T3)
    #0 0x43f8d7 in ucompthread ../stream.c:1523
    #1 0x7f81218fc6b9 in start_thread (/lib/x86_64-linux-gnu/libpthread.so.0+0x76b9)
    #2 0x7f8120d2e41c in clone (/lib/x86_64-linux-gnu/libc.so.6+0x10741c)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV ../stream.c:1523 ucompthread
Thread T3 created by T0 here:
    #0 0x7f81221941e3 in pthread_create (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x361e3)
    #1 0x4516f3 in create_pthread ../stream.c:133
    #2 0x4516f3 in fill_buffer ../stream.c:1699
    #3 0x4516f3 in read_stream ../stream.c:1786

==161258==ABORTING
```

5hadowblad3 changed the title ~~Segmentation fault caused by null pointer deference during multithread processing in ucompthread, stream.c:1523~~ Segmentation fault caused by null pointer deference during multithread processing in ucompthread, stream.c:1523 on Sep 2, 2020

5hadowblad3 changed the title ~~Segmentation fault caused by null pointer deference during multithread processing in ucompthread, stream.c:1523~~ Segmentation fault caused by null pointer dereference during multithread processing in ucompthread, stream.c:1523 on Sep 2, 2020

pete4abw commented on Sep 3, 2020 Contributor

Nope. A curious thing about `lrzip` is it requires a file extension. Testing a file without an extension has proven problematic. In any event, a properly named file works as expected even with your distractions and intentional munging. As I said, there is no way to account every act of intentional sabotage. Your file has an expected size of **70,506,183,141,503**. Enjoy the program. It works great.

```
peter@tommyv:~/Downloads$ lrzip.631 -tvv seg-stream1523.lrz
Using configuration file /home/peter/.lrzip/lrzip.conf
Threading is ENABLED. Number of CPUs detected: 8
Detected 16563281920 bytes ram
Compression level 7
Nice Value: 19
Show Progress
```

```
Max Verbose
Test file integrity
Temporary Directory set as: ./
Detected lrzip version 0.6 file.
Unknown hash, falling back to CRC
CRC32 being used for integrity testing.
Decompressing...
Reading chunk_bytes at 24
Expected size: 70506183141503
Chunk byte width: 2
Reading eof flag at 25
EOF: 1
Reading expected chunksize at 26
Chunk size: 10240
Reading stream 0 header at 29
Reading stream 1 header at 36
Reading ucomp header at 43
Fill_buffer stream 0 c_len 55 u_len 55 last_head 0
Starting thread 0 to decompress 55 bytes from stream 0
Thread 0 decompressed 55 bytes from stream 0
Taking decompressed data from thread 0
Reading ucomp header at 105
Fill_buffer stream 1 c_len 269 u_len 9387 last_head 131
Starting thread 1 to decompress 269 bytes from stream 1
Reading ucomp header at 160
Fill_buffer stream 1 c_len 24 u_len 985 last_head 0
Thread 1 decompressed 9387 bytes from stream 1
Starting thread 2 to decompress 24 bytes from stream 1
Taking decompressed data from thread 1
Closing stream at 190, want to seek to 411
Bad checksum: 0x5b496f91 - expected: 0x2000210c
Fatal error - exiting



peter@tommyv:~/Downloads$ lrzip.631 -ivv seg-stream1523.lrz
Using configuration file /home/peter/.lrzip/lrzip.conf
Detected lrzip version 0.6 file.
Unknown hash, falling back to CRC
Rzip chunk 1:
Chunk byte width: 2
Chunk size: 10240
Stream: 0
Offset: 28
Block   Comp    Percent Size
1       none    100.0%  55 / 55 Offset: 0       Head: 0
Stream: 1
Offset: 28
Block   Comp    Percent Size
1       none    2.9%    269 / 9387      Offset: 0       Head: 131
2       lzma    2.7%    24 / 985        Offset: 0       Head: 0
Invalid chunk bytes 20
No such file or directory
Fatal error - exiting
```

5hadowblad3 mentioned this issue on Sep 4, 2020

Segmentation fault casued by use after free in multithread process from close_stream_in, stream:1870 to lzma_decompress_buf, stream:546 #165

Closed

5hadowblad3 commented on Sep 4, 2020 • edited — Author

Well, since it is a multithread issue, you still can use the uploaded file (without adding an extension name) to reproduce this segmentation fault by running the command multiple times. I add a more detailed explanation related to this bug in the newest issue #165 for another related bug.

pete4abw mentioned this issue on Oct 29, 2020

Fixes to Corrupt File errors and segfaults #171

Closed

ckolivas commented on Feb 14, 2021 — Owner

Fixed in git.

ckolivas closed this as completed on Feb 14, 2021

Clingto mentioned this issue on May 19, 2021

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

Closed

5hadowblad3 commented on Jun 9, 2021 — Author

This is assigned with CVE-2021-27345.

pete4abw mentioned this issue on Jun 9, 2021

CVEs 2020-25467 and 2021-27345 and 2021-27347 pete4abw/lrzip-next#30

Closed

carnil commented on Apr 9

Fixing ocmmit should be `be884d0`

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants