appneta / tcpreplay Public

Code | Issues 29 | Pull requests 5 | Actions | Projects 6 | Wiki | ...

New issue

Jump to bottom

[Bug] heap-overflow in get_l2len_protocol #716

Closed kdsjZh opened this issue on Mar 3 · 2 comments

Assignees

Projects 4.4.2

kdsjZh commented on Mar 3 • edited

You are opening a *bug report* against the Tcpreplay project: we use GitHub Issues for tracking bug reports and feature requests.

If you have a question about how to use Tcpreplay, you are at the wrong site. You can ask a question on the tcpreplay-users mailing list or on Stack Overflow with [tcpreplay] tag.
General help is available here.

If you have a build issue, consider downloading the latest release

Otherwise, to report a bug, please fill out the reproduction steps (below) and delete these introductory paragraphs. Thanks!

Describe the bug
There is a heap-overflow bug found in get_l2len_protocol, can be triggered via tcpprep + ASan

To Reproduce
Steps to reproduce the behavior:

1. export CC=clang
2. export CFLAGS="-fsanitize=address -g"
3. ./autogen.sh && ./configure --disable-shared --disable-local-libopts && make clean && make -j8
4. ./src/tcpprep --auto=bridge --pcap=$POC --cachefile=/dev/null

Expected behavior
ASan report that ./tcpprep has a heap buffer overflow in function get_l2len_protocol

```
Warning: crash.0 was captured using a snaplen of 1 bytes.  This may mean you have truncated
packets.
=================================================================
==22937==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60200000001c at pc
0x000000510fb4 bp 0x7ffd68b94250 sp 0x7ffd68b94248
READ of size 2 at 0x60200000001c thread T0
    #0 0x510fb3 in get_l2len_protocol /benchmark/vulnerable/tcpreplay/src/common/get.c:322:46
    #1 0x512222 in get_ipv4 /benchmark/vulnerable/tcpreplay/src/common/get.c:442:11
    #2 0x4f82f2 in process_raw_packets /benchmark/vulnerable/tcpreplay/src/tcpprep.c:368:41
    #3 0x4f7929 in main /benchmark/vulnerable/tcpreplay/src/tcpprep.c:144:23
    #4 0x7fc5856d2bf6 in __libc_start_main /build/glibc-S9d2JN/glibc-2.27/csu/../csu/libc-
start.c:310
    #5 0x41c1b9 in _start (/benchmark/vulnerable/tcpreplay/src/tcpprep+0x41c1b9)

0x60200000001c is located 11 bytes to the right of 1-byte region [0x602000000010,0x602000000011)
allocated by thread T0 here:
    #0 0x4aeb80 in malloc /home/nipc/workspace/install/llvm-project/llvm/projects/compiler-
rt/lib/asan/asan_malloc_linux.cpp:145
    #1 0x7fc586add90f  (/usr/lib/x86_64-linux-gnu/libpcap.so.0.8+0x1f90f)

SUMMARY: AddressSanitizer: heap-buffer-overflow
/benchmark/vulnerable/tcpreplay/src/common/get.c:322:46 in get_l2len_protocol
Shadow bytes around the buggy address:
  0x0c047fff7fb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c047fff8000: fa fa 01[fa]fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8010: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8050: 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
==22937==ABORTING
```

Screenshots

```
Warning: crash.0 was captured using a snaplen of 1 bytes.  This may mean you have truncated packets.
=================================================================
==22937==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60200000001c at pc 0x000000510fb4 bp 0x7ffd68b94250 sp 0x7ffd68b94248
READ of size 2 at 0x60200000001c thread T0
    #0 0x510fb3 in get_l2len_protocol /benchmark/vulnerable/tcpreplay/src/common/get.c:322:46
    #1 0x512222 in get_ipv4 /benchmark/vulnerable/tcpreplay/src/common/get.c:442:11
    #2 0x4f82f2 in process_raw_packets /benchmark/vulnerable/tcpreplay/src/tcpprep.c:368:41
    #3 0x4f7929 in main /benchmark/vulnerable/tcpreplay/src/tcpprep.c:144:23
    #4 0x7fc5856d2bf6 in __libc_start_main /build/glibc-S9d2JN/glibc-2.27/csu/../csu/libc-start.c:310
    #5 0x41c1b9 in _start (/benchmark/vulnerable/tcpreplay/src/tcpprep+0x41c1b9)

0x60200000001c is located 11 bytes to the right of 1-byte region [0x602000000010,0x602000000011)
allocated by thread T0 here:
    #0 0x4aeb80 in malloc /home/nipc/workspace/install/llvm-project/llvm/projects/compiler-rt/lib/asan/asan_malloc_linux.cpp:145
    #1 0x7fc586add90f  (/usr/lib/x86_64-linux-gnu/libpcap.so.0.8+0x1f90f)

SUMMARY: AddressSanitizer: heap-buffer-overflow /benchmark/vulnerable/tcpreplay/src/common/get.c:322:46 in get_l2len_protocol
Shadow bytes around the buggy address:
  0x0c047fff7fb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c047fff8000: fa fa 01[fa]fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8010: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8050: 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
```

System (please complete the following information):

- OS: Ubuntu
- (can be reproduced in 20.04 & 18.04)
- Tcpreplay Version (latest commit `09f0774`)

Credit
Han Zheng
NCNIPC of China
Hexhive

kdsjZh commented on Mar 3 Author

POC.zip

fklassen added this to **To do** in **4.4.2** on Apr 22

chluo911 mentioned this issue on Jul 24

[Bug] heap-overflow in get.c:344 #735

Closed

fklassen self-assigned this on Aug 1

fklassen moved this from **To do** to **In progress** in **4.4.2** on Aug 1

fklassen added a commit that referenced this issue on Aug 1

`But #716 heap-buffer-overflow in get_l2len_protocol()` `a135cea`

fklassen added a commit that referenced this issue on Aug 1

`Merge pull request #738 from appneta/Bug_#716_tcpreplay_heap-buffer-o…` ··· `43622a5`

fklassen commented on Aug 1 Member

Fixed in PR #738.

Must check that ether size is at least 14 bytes long before parsing.

fklassen closed this as completed on Aug 1

4.4.2 automation moved this from **In progress** to **Done** on Aug 1

fklassen mentioned this issue on Aug 1

[Bug] Reachable assertion in packet2tree() #715

Closed

Assignees

fklassen

Labels

None yet

Projects

4.4.2
Done

Milestone

No milestone

Development

No branches or pull requests

2 participants