

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

# [Bug] heap buffer overflow in parse\_mpls #719

✓ Closed

kdsjZh opened this issue on Mar 14 · 1 comment

Projects

 4.4.2

kdsjZh commented on Mar 14 • 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 parse\_mpls, can be triggered via tcpprep+ ASan

## To Reproduce

Steps to reproduce the behavior:

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

Output:

```
==2021941==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000142 at pc
0x000000448721 bp 0x7fff4fd006f0 sp 0x7fff4fd006e0
READ of size 4 at 0x602000000142 thread T0
#0 0x448720 in parse_mpls (/validate/run1/tcpreplay/tcpprep+0x448720)
#1 0x44edb0 in parse_metadata (/validate/run1/tcpreplay/tcpprep+0x44edb0)
```

```
#2 0x44c591 in get_l2len_protocol (/validate/run1/tcpreplay/tcpprep+0x44c591)
#3 0x44fa30 in get_ipv4 (/validate/run1/tcpreplay/tcpprep+0x44fa30)
#4 0x41434c in process_raw_packets (/validate/run1/tcpreplay/tcpprep+0x41434c)
#5 0x412708 in main (/validate/run1/tcpreplay/tcpprep+0x412708)
#6 0x7f6b96777d8f (/lib/x86_64-linux-gnu/libc.so.6+0x2dd8f)
#7 0x7f6b96777e3f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2de3f)
#8 0x4085f4 in _start (/validate/run1/tcpreplay/tcpprep+0x4085f4)
```

0x602000000142 is located 2 bytes to the right of 16-byte region [0x602000000130,0x602000000140) allocated by thread T0 here:

```
#0 0x4dd0d8 in __interceptor_realloc (/validate/run1/tcpreplay/tcpprep+0x4dd0d8)
#1 0x7f6b969bd1c7 (/lib/x86_64-linux-gnu/libpcap.so.0.8+0x291c7)
```

SUMMARY: AddressSanitizer: heap-buffer-overflow (/validate/run1/tcpreplay/tcpprep+0x448720) in parse\_mpls

Shadow bytes around the buggy address:

```
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 fd fd fa fa fd fd fa fa fd fd fa fa fd fd
0x0c047fff8010: fa fa fd fd fa fa fd fd fa fa fd fd fa fa fd fd
=>0x0c047fff8020: fa fa fd fd fa fa 00 00[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
0x0c047fff8060: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c047fff8070: 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
```

==2021941==ABORTING

System (please complete the following information):

- OS: Ubuntu 20.04
- Clang 12.0.1
- Tcpreplay Version : latest commit [09f0774](#)



## Credit

[NCNIPC of China](#)

[Hexhive](#)

## POC

[POC2.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:150 #736**

 Closed

**fklassen** commented on Aug 4

Member

Improved overflow protection for parse\_mpls() in PR [#741](#)

 **fklassen** closed this as completed on Aug 4

 **4.4.2**  moved this from **To do** to **Done** on Aug 4

 **fklassen** added a commit that referenced this issue on Aug 26

 Bug [#719](#) better overflow protection in parse\_mpls

54297ef

 **fklassen** added a commit that referenced this issue on Aug 26

 Merge pull request [#741](#) from appneta/Bug\_#719\_heap-overflow\_in\_parse\_... ...

1de1a21

## Assignees

No one assigned

## Labels

None yet

Projects



4.4.2

Done

Milestone

No milestone

Development

No branches or pull requests

2 participants

