

New issue

Jump to bottom

[Bug]heap buffer overflow in tcpprep with get_l2len() #617

Closed

jimoyong opened this issue on Jul 30, 2020 · 3 comments

Assignees



Labels

bug

Projects

4.3.4

jimoyong commented on Jul 30, 2020 • edited

Describe the bug

A heap buffer overflow found in tcpprep with get_l2len().

ASAN report:

```
==83==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000034 at pc 0x0000004e1900 bp 0x7fffd14c660 sp 0x7fffd14c658
READ of size 2 at 0x602000000034 thread T0
#0 0x4e18ff in get_l2len /src/tcpplay-4.3.3/src/common/get.c:191:22
#1 0x4e1b2b in get_ipv4 /src/tcpplay-4.3.3/src/common/get.c:267:14
#2 0x4c8c99 in process_raw_packets /src/tcpplay-4.3.3/src/tcpplay.c:370:41
#3 0x4c8c99 in main /src/tcpplay-4.3.3/src/tcpplay.c:147:23
#4 0x7f3c98e6683f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2083f)
#5 0x41c348 in _start (/out/tcpplay+0x41c348) _### //I just rename tcpprep to tcpplay//_

0x602000000034 is located 0 bytes to the right of 4-byte region [0x602000000030,0x602000000034)
allocated by thread T0 here:
#0 0x49619d in malloc (/out/tcpplay+0x49619d)
#1 0x7f3c99f904fe (/usr/lib/x86_64-linux-gnu/libc.so.0.8+0x1f4fe)

SUMMARY: AddressSanitizer: heap-buffer-overflow /src/tcpplay-4.3.3/src/common/get.c:191:22 in get_l2len
Shadow bytes around the buggy address:
 0x0c047fff7fb0: 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
 0x0c047fff7fd0: 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
 0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c047fff8000: fa fa 00 03 fa fa[04]fa fa fa fa fa fa fa fa
0x0c047fff8010: 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
0x0c047fff8030: 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
0x0c047fff8050: 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
==83==ABORTING
```

To Reproduce

Steps to reproduce the behavior:

1. download tcpplay-4.3.3.tar.gz
2. apt-get -y install libpcap-dev
3. cd tcpplay-3.4.4 && ./configure && make && make install
4. tcpprep -a client -i [poc_filename] -o a.cach

[poc_tcpplay_heap_buffer_overflow_get_l2len.tar.gz](#)

Expected behavior

Get an a.cach at the path or exit when meet abnormal input.

System (please complete the following information):

- Tcpplay Version 4.3.3 tcpprep -V



```
tcpprep version: 4.3.3 (build git:v4.3.3)
Copyright 2013-2018 by Fred Klassen <tcpplay at appneta dot com> - AppMeta
Copyright 2000-2012 by Aaron Turner <aturner at synfin dot net>
The entire Tcpplay Suite is licensed under the GPLv3
Cache file supported: 04
Not compiled with libndnet.
Compiled against libpcap: 1.7.4
64 bit packet counters: enabled
```

Verbose printing via tcpdump: disabled

OS: ubuntu-16.04.6 x86_64

Additional context
None.

 **GabrielGanne** added a commit to GabrielGanne/tcpreplay that referenced this issue on Aug 3, 2020

 **fix heap-buffer-overflow when DLT_JUNIPER_ETHER** ... 

d311085

  **GabrielGanne** mentioned this issue on Aug 3, 2020

fix heap-buffer-overflow when DLT_JUNIPER_ETHER #618

 Merged



  **fklassen** self-assigned this on Aug 3, 2020

  **fklassen** added the `bug` label on Aug 3, 2020

  **fklassen** added this to To do in 4.3.4 via `automation` on Aug 3, 2020

carnil commented on Oct 23, 2020

[CVE-2020-24266](#) got assigned for this issue.

  **cbiedl** mentioned this issue on Dec 19, 2020

[Bug]heap-buffer-overflow in tcpprep with MemcmpInterceptorCommon() #616

 Closed

  **dotlambda** mentioned this issue on Feb 2, 2021

Vulnerability roundup 96: tcpreplay-4.3.3: 2 advisories [7.5] NixOS/nixpkgs#102902

 Closed

 2 tasks

 **fklassen** added a commit that referenced this issue on Mar 12, 2021

 Merge pull request [#638](#) from appneta/Bug_#617_CVE-2020-24266 ...


765f012


fklassen commented on Mar 12, 2021

Member

Fixed in [#616](#) and PR [#637](#). PR [#638](#) updates changelog.

 **fklassen** closed this as completed on Mar 12, 2021

 4.3.4 `automation` moved this from To do to Done on Mar 12, 2021

 **fklassen** added a commit that referenced this issue on Mar 13, 2021

 Bug [#620](#) apply get.c functions fixed in [#617](#) ...

21cce6b

 **fklassen** added a commit that referenced this issue on Mar 13, 2021

 Merge pull request [#640](#) from appneta/Bug_#620_heap-buffer-overflow_wi_ ...

1541a39

  **fklassen** mentioned this issue on Mar 13, 2021

[Bug] heap-buffer-overflow in tcpreplay with fast_edit_packet() #620

 Closed

 **fklassen** added a commit that referenced this issue on Mar 13, 2021

 Bug [#620](#) apply get.c functions fixed in [#617](#) ...

61db8ad

fklassen commented on Aug 25, 2021

Member

From mail lists:

Hi,

The following vulnerability was published for tcpreplay.

CVE-2020-24266[0]:

| An issue was discovered in tcpreplay tcpprep v4.3.3. There is a heap
| buffer overflow vulnerability in get_l2len() that can make tcpprep
| crash and cause a denial of service.

If you fix the vulnerability please also make sure to include the
CVE (Common Vulnerabilities & Exposures) id in your changelog entry.

For further information see:

[0] <https://security-tracker.debian.org/tracker/CVE-2020-24266>
<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-24266>

[1] [#617](#)

Please adjust the affected versions in the BTS as needed.

Regards,
Salvatore

Assignees

 fklassen

Labels

bug

Projects

No open projects

1 closed project ▾

Milestone

No milestone

Development

No branches or pull requests

3 participants

