New issue

Jump to bottom

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

Closed jimoyong opened this issue on Jul 30, 2020 · 9 comments

Assignees

Labels bug

Projects 4.3.4

jimoyong commented on Jul 30, 2020 • edited

What's the problem (or question)?
A heap buffer overflow with MemcmpInterceptorCommon() in the 4.3.3 version of tcpprep.

```
==74==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000032 at pc 0x000000432f16 bp 0x7ffe3a489250 sp 0x7ffe3a4889f8
READ of size 3 at 0x602000000032 thread T0
    #0 0x432f15 in MemcmpInterceptorCommon(void*, int (*)(void const*, void const*, unsigned long), void const*, void const*, unsigned long) (/out/tcpreplay+0x432f15)
    #1 0x43346a in bcmp (/out/tcpreplay+0x43346a)
    #2 0x4e1513 in get_l2len /src/tcpreplay-4.3.3/src/common/get.c:186:13
    #3 0x4e1b2b in get_ipv4 /src/tcpreplay-4.3.3/src/common/get.c:267:14
    #4 0x4c8c99 in process_raw_packets /src/tcpreplay-4.3.3/src/tcpprep.c:370:41
    #5 0x4c8c99 in main /src/tcpreplay-4.3.3/src/tcpprep.c:147:23
    #6 0x7f97e73b883f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2083f)
    #7 0x41c348 in _start (/out/tcpreplay+0x41c348) //I just rename tcpprep to tcpreplay//

0x602000000032 is located 0 bytes to the right of 2-byte region [0x602000000030,0x602000000032)
allocated by thread T0 here:
    #0 0x49619d in malloc (/out/tcpreplay+0x49619d)
    #1 0x7f97e84e24fe  (/usr/lib/x86_64-linux-gnu/libpcap.so.0.8+0x1f4fe)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/out/tcpreplay+0x432f15) in MemcmpInterceptorCommon(void*, int (*)(void const*, void const*, unsigned long), void const*, void
const*, unsigned long)
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 00 03 fa fa[02]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
==74==ABORTING
```

Steps to reproduce the behavior:

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

poc_tcpprep_heap_buffer_overflow_MemcmpInterceptorCommon.tar.gz

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

System :

- Tcpreplay Version 4.3.3 tcpprep -V

```
Copyright 2013-2018 by Fred Klassen <tcpreplay at appneta dot com> - AppNeta
Copyright 2000-2012 by Aaron Turner <aturner at synfin dot net>
The entire Tcpreplay Suite is licensed under the GPLv3
Cache file supported: 04
Not compiled with libdnet.
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 this to **To do** in **4.3.4** via automation on Aug 3, 2020

fklassen added the bug label on Aug 3, 2020

carnil commented on Oct 23, 2020

CVE-2020-24265 go assigned for this issue.

cbiedl commented on Dec 19, 2020

Excuse my bluntness, wouldn't it be sufficient to check whether there's enough data before accessing it, in other words:

```
          if ((pktdata[3] & 0x80) == 0x80) {
+             if (datalen < 6)
+                 errx(-1, "short packet data (%u)", datalen);
              l2_len = ntohs(*((uint16_t*)&pktdata[4]));
              l2_len += 6;
          } else {
```

cbiedl commented on Dec 19, 2020

Ooops, the previous comment should have gone to #617

For this one, suggestion is likewise:

```
+         if (datalen < 4)
+             errx(-1, "short packet data (%u)", datalen);
          if (memcmp(pktdata, "MGC", 3))
              warnx("No Magic Number found: %s (0x%x)",
                   pcap_datalink_val_to_description(datalink), datalink);
```

Did I miss the point?

fklassen commented on Jan 11, 2021 Member

> Ooops, the previous comment should have gone to #617
>
> For this one, suggestion is likewise:
>
> ```
> +         if (datalen < 4)
> +             errx(-1, "short packet data (%u)", datalen);
>           if (memcmp(pktdata, "MGC", 3))
>               warnx("No Magic Number found: %s (0x%x)",
>                    pcap_datalink_val_to_description(datalink), datalink);
> ```
>
> Did I miss the point?

@cbiedl thanks for the feedback. I plan to schedule some time in the next 2 weeks to address this and other issues.

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

dotlambda commented on Feb 3, 2021

> I plan to schedule some time in the next 2 weeks to address this and other issues.

@fklassen Any updates?

fklassen moved this from **To do** to **In progress** in **4.3.4** on Mar 12, 2021

fklassen commented on Mar 12, 2021 Member

> I plan to schedule some time in the next 2 weeks to address this and other issues.
>
> **@fklassen** Any updates?

Had a backlog of work so took some vacation to address this and a few other critical bugs.

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

`Bug #616 add checks for datalen for DLT_JUNIPER_ETHER` ··· `8323a7f`

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

`Merge pull request #637 from appneta/Bug_#616_CVE-2020-24265` ··· `6fb578d`

fklassen commented on Mar 12, 2021 Member

Adde checks for `datalen` for DLT_JUNIPER_ETHER

Also did some fixes to Juniper Ethernet protocols to fix some bugs
and support various types of Juniper Ethernet protocol types. Used
Wireshark sources to figure out all the different packet types that
Juniper uses.

Unable to test all types because of lack of JNPER DLT pcaps.

Also applied a fix for DLT_RAW to prevent similar issues.

fklassen closed this as completed on Mar 12, 2021

4.3.4 automation moved this from **In progress** to **Done** on Mar 12, 2021

fklassen commented on Mar 12, 2021 Member

> Excuse my bluntness, wouldn't it be sufficient to check whether there's enough data before accessing it, in other words:
>
> ```
>          if ((pktdata[3] & 0x80) == 0x80) {
> +            if (datalen < 6)
> +                errx(-1, "short packet data (%u)", datalen);
>              l2_len = ntohs(*((uint16_t*)&pktdata[4]));
>              l2_len += 6;
>          } else {
> ```

Bluntness excepted. Juniper protocol has very little testing available because I have a lack of PCAP files available. I have never seen the PCAP files that were used to create this feature. I am fixing base on looking at Wireshark decodes.

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

`Bug_#617_CVE-2020-24266 fixed by #616` ··· `f86f2cd`

This was referenced on Mar 12, 2021

Bug #617 CVE-2020-24266 fix tcpprep get_l2len() #638

Merged

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

Closed

dumprop mentioned this issue on May 9, 2021

[Bug]heap-buffer-overflow with flow_decode() #665

Closed

fklassen added a commit that referenced this issue on Jun 19, 2021

`Merge pull request #666 from dumprop/master` ··· `9f6f3d5`

fklassen commented on Aug 25, 2021 Member

From mail lists:

Hi,

The following vulnerability was published for tcpreplay.

CVE-2020-24265[0]:
| An issue was discovered in tcpreplay tcpprep v4.3.3. There is a heap
| buffer overflow vulnerability in MemcmpInterceptorCommon() 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-2020-24265
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-24265
[1] #616

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

5 participants