[Bug] tcpreplay-edit ——heap-buffer-overflow in randomize_iparp at edit_packet.c:1032 #579

Closed **14isnot40** opened this issue on May 19, 2020 · 2 comments

Assignees	
Labels	bug
Projects	4.3.3
Milestone	4.3.3

14isnot40 commented on May 19, 2020

Describe the bug
A heap-based buffer overflow was discovered in tcpreplay-edit binary, during the pointer 'ip' dereference operation. The issue is being triggered in the function randomize_iparp at edit_packet.c:1032.

To Reproduce
Steps to reproduce the behavior:

1. Compile tcpreplay according to the default configuration

```
./configure CFLAGS="-g -O0 -fsanitize=address"
```

2. execute command

```
tcpreplay-edit -r 80:84 -s 20 -b -C -m 1500 -P --oneatatime -i lo $poc
```

poc can be found here.

Expected behavior
An attacker can exploit this vulnerability by submitting a malicious pcap that exploits this issue. This will result in a Denial of Service (DoS), potentially Information Exposure when the application attempts to process the file.

Screenshots
ASAN Reports

```
==64974==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60300000edf6 at pc 0x000000425341 bp 0x7fffffffd5d0 sp 0x7fffffffd5c0
READ of size 4 at 0x60300000edf6 thread T0
    #0 0x425340 in randomize_iparp /home/test/Desktop/evaulation/tcpreplay/src/tcpedit/edit_packet.c:1032
    #1 0x41c71b in tcpedit_packet /home/test/Desktop/evaulation/tcpreplay/src/tcpedit/tcpedit.c:329
    #2 0x40963b in send_packets /home/test/Desktop/evaulation/tcpreplay/src/send_packets.c:552
    #3 0x418e9a in replay_file /home/test/Desktop/evaulation/tcpreplay/src/replay.c:182
    #4 0x417e73 in tcpr_replay_index /home/test/Desktop/evaulation/tcpreplay/src/replay.c:59
    #5 0x416de4 in tcpreplay_replay /home/test/Desktop/evaulation/tcpreplay/src/tcpreplay_api.c:1136
    #6 0x40fb4f in main /home/test/Desktop/evaulation/tcpreplay/src/tcpreplay.c:139
    #7 0x7ffff687f82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #8 0x403508 in _start (/usr/local/bin/tcpreplay-edit+0x403508)

0x60300000edf6 is located 6 bytes to the right of 32-byte region [0x60300000edd0,0x60300000edf0)
allocated by thread T0 here:
    #0 0x7ffff6f02602 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x98602)
    #1 0x7ffff6c484fe  (/usr/lib/x86_64-linux-gnu/libpcap.so.0.8+0x1f4fe)

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/test/Desktop/evaulation/tcpreplay/src/tcpedit/edit_packet.c:1032 randomize_iparp
Shadow bytes around the buggy address:
  0x0c067fff9d60: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c067fff9d70: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c067fff9d80: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c067fff9d90: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c067fff9da0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x0c067fff9db0: fa fa fa fa fa fa fa fa fa fa 00 00 00 00[fa]fa
  0x0c067fff9dc0: 00 00 00 fa fa fa fd fd fd fa fa fa fa fd fd fd fa
  0x0c067fff9dd0: fa fa fd fd fd fd fd fa fa fd fd fd fd fa fa fd fd
  0x0c067fff9de0: fd fd fa fa fd fd fd fd fd fa fa fd fd fd fa fa fa
  0x0c067fff9df0: fd fd fd fa fa fa fd fd fd fd fd fa fa fd fd fd fa
  0x0c067fff9e00: 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
  Heap right redzone:      fb
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack partial redzone:   f4
  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
==64974==ABORTING
```

Debug

```
gef➤  ni
0x0000000000425339      1032            *ip = randomize_ipv4_addr(tcpedit, *ip);
[ Legend: Modified register | Code | Heap | Stack | String ]
──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────── registers ────
$rax   : 0x000060300000edf6  →  0x00010000001802ff
$rbx   : 0x00007fffffffd800  →  0x00007fffffffdb80  →  0x00000fffffffb7a  →  0x0000000000000000
$rcx   : 0x1
$rdx   : 0x1
$rsp   : 0x00007fffffffd5e0  →  0x00000001ffffd620  →  0x0000000000000000
$rbp   : 0x00007fffffffd620  →  0x00007fffffffd820  →  0x00007fffffffdba0  →  0x00007fffffffdd60  →  0x00007fffffffdd90  →  0x00007fffffffde90  →  0x00007fffffffe340  →  0x00000000004
$rsi   : 0x9
$rdi   : 0x100
$rip   : 0x0000000000425339  →  <randomize_iparp+625> mov rdi, rax
$r8    : 0x7
$r9    : 0x12018001fffff86
$r10   : 0x895
$r11   : 0x00007ffff69783a0  →  <ntohs+0> mov eax, edi
$r12   : 0x00000fffffffffad4  →  0x0000000000000000
$r13   : 0x00007fffffffd6a0  →  0x0000000041b58ab3
$r14   : 0x00007fffffffd6a0  →  0x0000000041b58ab3
$r15   : 0x00007fffffffdbd0  →  0x0000000041b58ab3
$eflags: [carry parity adjust zero sign trap INTERRUPT direction overflow resume virtualx86 identification]
$cs: 0x0033 $ss: 0x002b $ds: 0x0000 $es: 0x0000 $fs: 0x0000 $gs: 0x0000
──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────── stack ────
0x00007fffffffd5e0│+0x0000: 0x00000001ffffd620  →  0x0000000000000000    ← $rsp
0x00007fffffffd5e8│+0x0008: 0x000060300000edd0  →  0x00eb5000e8000000
0x00007fffffffd5f0│+0x0010: 0x00007fffffffdac0  →  0x00000000546031b8
0x00007fffffffd5f8│+0x0018: 0x000061d00001ea80  →  0x0000000000000001
0x00007fffffffd600│+0x0020: 0x0000000effffd800  →  0x0000000000000000
0x00007fffffffd608│+0x0028: 0x000060300000edde  →  0x0100001000080062 ("b"?)
0x00007fffffffd610│+0x0030: 0x000060300000edf6  →  0x00010000001802ff
0x00007fffffffd618│+0x0038: 0x000060300000edf6  →  0x00010000001802ff
──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────── code:x86:64 ────
     0x425332 <randomize_iparp+618> ret    0xca21
     0x425335 <randomize_iparp+621> test   dl, dl
     0x425337 <randomize_iparp+623> je     0x425341 <randomize_iparp+633>
 →   0x425339 <randomize_iparp+625> mov    rdi, rax
     0x42533c <randomize_iparp+628> call   0x402ba0 <__asan_report_load4@plt>
     0x425341 <randomize_iparp+633> mov    rax, QWORD PTR [rbp-0x8]
     0x425345 <randomize_iparp+637> mov    edx, DWORD PTR [rax]
     0x425347 <randomize_iparp+639> mov    rax, QWORD PTR [rbp-0x28]
     0x42534b <randomize_iparp+643> mov    esi, edx
──────────────────────────────────────────────────────────────────────────────────────────────────────────────────── source:edit_packet.c+1032 ────
   1027          memcpy(&iptemp, add_hdr, sizeof(uint32_t));
   1028          ip = &iptemp;
   1029  #else
   1030          ip = (uint32_t *)add_hdr;
   1031  #endif
                 // ip=0x00007fffffffd618  →  [...]  →  0x00010000001802ff, tcpedit=0x00007fffffffd5f8  →  [...]  →  0x0000000000000001
 → 1032          *ip = randomize_ipv4_addr(tcpedit, *ip);
   1033  #ifdef FORCE_ALIGN
   1034          memcpy(add_hdr, &iptemp, sizeof(uint32_t));
   1035  #endif
   1036
   1037          add_hdr += arp_hdr->ar_pln + arp_hdr->ar_hln;
──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────── threads ────
[#0] Id 1, Name: "tcpreplay-edit", stopped, reason: SINGLE STEP
──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────── trace ────
[#0] 0x425339 → randomize_iparp(tcpedit=0x61d00001ea80, pkthdr=0x7fffffffdac0, pktdata=0x60300000edd0 "", datalink=0x1)
[#1] 0x41c71c → tcpedit_packet(tcpedit=0x61d00001ea80, pkthdr=0x7fffffffd940, pktdata=0x7fffffffd8c0, direction=TCPR_DIR_C2S)
[#2] 0x40963c → send_packets(ctx=0x61e00000f080, pcap=0x61600000f380, idx=0x0)
[#3] 0x418e9b → replay_file(ctx=0x61e00000f080, idx=0x0)
[#4] 0x417e74 → tcpr_replay_index(ctx=0x61e00000f080)
[#5] 0x416de5 → tcpreplay_replay(ctx=0x61e00000f080)
[#6] 0x40fb50 → main(argc=0x1, argv=0x7fffffffe490)
────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
gef➤  p ip
$3 = (uint32_t *) 0x60300000edf6
gef➤  p *ip
$4 = 0x1802ff
```

System (please complete the following information):

- OS version : Ubuntu 16.04
- Tcpreplay Version : 4.3.2/master branch

fklassen self-assigned this on May 20, 2020

fklassen added the bug label on May 20, 2020

fklassen added this to the **4.3.3** milestone on May 20, 2020

fklassen commented on Jun 1, 2020 Member

I am unable to playback the poc file. It also gets the same failure if I attempt to open with wireshark or tcpdump.

```
PID: 128257
Warning in sendpacket.c:sendpacket_open_pf() line 943:
Unsupported physical layer type 0x0304 on lo.  Maybe it works, maybe it won't.  See tickets #123/318

Failed: From replay.c:replay_file() line 129:
Error opening pcap file: unsupported pcap savefile version 2.250
```

fklassen added this to **To do** in **4.3.3** via automation on Jun 1, 2020

fklassen moved this from **To do** to **In progress** in **4.3.3** on Jun 1, 2020

fklassen commented on Jun 1, 2020 Member

Addressed issue based on reported stack trace. Unable to reproduce so unable to verify. All tests passed.

Fixed in PR #588

fklassen closed this as completed on Jun 1, 2020

4.3.3 automation moved this from **In progress** to **Done** on Jun 1, 2020

Assignees

fklassen

Labels

bug

Projects

No open projects

1 closed project

Milestone

4.3.3

Development

No branches or pull requests

2 participants