tcprewrite —— heap-buffer-overflow in get_ipv6_next()

Describe the bug

A heap-based buffer overflow was discovered in tcprewrite binary, during the get_c operation. The issue is being triggered in the function get_ipv6_next() at common/get.c.

To Reproduce

Steps to reproduce the behavior:

1. Compile tcpreplay according to the default configuration
2. execute command

```
tcprewrite -i $poc -o /dev/null --fuzz-seed=42
```

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) and potentially Information Exposure when the application attempts to process the file.

Screenshots

ASAN Reports

```
/usr/local/bin/tcprewrite -i id\:000000\,sig\:11\,src\:000280\,op\:fa-havoc\,rep\:2 -o /dev/null --fuzz-seed=42
=================================================================
==34195==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x63100001080e at pc 0x00000042bd74 bp 0x7ffd8b9eada0 sp 0x7ffd8b9ead90
READ of size 4 at 0x63100001080e thread T0
```

```
    #0 0x42bd73 in get_ipv6_next /home/test/Desktop/evaulation/tcpreplay/src/common/get.c:454
    #1 0x42bfcc in get_ipv6_l4proto /home/test/Desktop/evaulation/tcpreplay/src/common/get.c:540
    #2 0x42bfb9 in get_ipv6_l4proto /home/test/Desktop/evaulation/tcpreplay/src/common/get.c:531
    #3 0x4134c2 in do_checksum /home/test/Desktop/evaulation/tcpreplay/src/tcpedit/checksum.c:63
    #4 0x40b383 in fix_ipv4_checksums /home/test/Desktop/evaulation/tcpreplay/src/tcpedit/edit_packet.c:74
    #5 0x4079c2 in tcpedit_packet /home/test/Desktop/evaulation/tcpreplay/src/tcpedit/tcpedit.c:354
    #6 0x40569b in rewrite_packets /home/test/Desktop/evaulation/tcpreplay/src/tcprewrite.c:291
    #7 0x404e13 in main /home/test/Desktop/evaulation/tcpreplay/src/tcprewrite.c:130
    #8 0x7f9fd6a0e82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #9 0x402688 in _start (/usr/local/bin/tcprewrite+0x402688)

0x63100001080e is located 1 bytes to the right of 65549-byte region [0x631000000800,0x63100001080d)
allocated by thread T0 here:
    #0 0x7f9fd72b2602 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x98602)
    #1 0x42c8e9 in _our_safe_malloc /home/test/Desktop/evaulation/tcpreplay/src/common/utils.c:50
    #2 0x40551e in rewrite_packets /home/test/Desktop/evaulation/tcpreplay/src/tcprewrite.c:249
    #3 0x404e13 in main /home/test/Desktop/evaulation/tcpreplay/src/tcprewrite.c:130
    #4 0x7f9fd6a0e82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/test/Desktop/evaulation/tcpreplay/src/common/get.c:454 get_ipv6_next
Shadow bytes around the buggy address:
  0x0c627fffa0b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c627fffa0c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c627fffa0d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
  0x0c627fffa0e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c627fffa0f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c627fffa100: 00[05]fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c627fffa110: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c627fffa120: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c627fffa130: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c627fffa140: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c627fffa150: 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
==34195==ABORTING
```

Debug

```

Program received signal SIGSEGV, Segmentation fault.
0x0000000000410025 in get_ipv6_next (exthdr=0x663ff6, len=0x8) at
get.c:454
454         maxlen = *((int*)((u_char *)exthdr + len));
[ Legend: Modified register | Code | Heap | Stack | String ]
──────────────────────────────────────────────────────────────────
──────────────────────────────────────────────────────────────────
registers ────
```

```
$rax   : 0x0000000000663ff6  →  0x0000000000000000
$rbx   : 0x0
$rcx   : 0x10080a0000000001
$rdx   : 0x1
$rsp   : 0x00007fffffffd8a8  →  0x0000000000410207  →
 <get_ipv6_l4proto+87> test rax, rax
$rbp   : 0x8
$rsi   : 0x8
$rdi   : 0x0000000000663ff6  →  0x0000000000000000
$rip   : 0x0000000000410025  →  <get_ipv6_next+37> mov esi, DWORD
PTR [rdi+rsi*1]
$r8    : 0xe
$r9    : 0x34
$r10   : 0x8
$r11   : 0x1
$r12   : 0x1008080000000001
$r13   : 0x1
$r14   : 0x20000000000
$r15   : 0x1
$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 ────
0x00007fffffffd8a8|+0x0000: 0x0000000000410207  →
 <get_ipv6_l4proto+87> test rax, rax     ← $rsp
0x00007fffffffd8b0|+0x0008: 0x0000000000633c4e  →
 0x29294fab8000a062 ("b"?)
0x00007fffffffd8b8|+0x0010: 0x0000000000631550  →
 0x0000000000000001
0x00007fffffffd8c0|+0x0018: 0x0000000000631550  →
 0x0000000000000001
0x00007fffffffd8c8|+0x0020: 0x000000000000000e
0x00007fffffffd8d0|+0x0028: 0x0000000000631550  →
 0x0000000000000001
0x00007fffffffd8d8|+0x0030: 0x0000000000406d56  →
 <do_checksum+438> mov ecx, DWORD PTR [rsp+0xc]
0x00007fffffffd8e0|+0x0038: 0x0000000000631e10  →
 0x0000000000631550  →  0x0000000000000001
```

```
──────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────────────────────────
code:x86:64 ────
     0x410014 <get_ipv6_next+20> add    BYTE PTR [rax+0x63], cl
     0x410017 <get_ipv6_next+23> test   BYTE PTR [rax-0x2d], 0xe2
     0x41001b <get_ipv6_next+27> movabs rcx, 0x10080a0000000001
 →   0x410025 <get_ipv6_next+37> mov    esi, DWORD PTR
[rdi+rsi*1]
     0x410028 <get_ipv6_next+40> test   rdx, rcx
     0x41002b <get_ipv6_next+43> jne    0x410050
<get_ipv6_next+80>
     0x41002d <get_ipv6_next+45> movabs rcx, 0x804000000000000
     0x410037 <get_ipv6_next+55> and    rcx, rdx
     0x41003a <get_ipv6_next+58> jne    0x410080
<get_ipv6_next+128>
```

System (please complete the following information):

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