New issue

Jump to bottom

A heap-based buffer overflow Read in RemoveSectionType in jpgfile.c #15

Closed **giantbranch** opened this issue on Feb 26, 2021 · 1 comment

giantbranch commented on Feb 26, 2021

Description of problem:

A heap-based buffer overflow Read in RemoveSectionType in jpgfile.c

Version-Release number of selected component (if applicable):

I tested the following version:
Jhead version: 3.05
Jhead version: 3.04

How reproducible:

git clone --depth=1 https://github.com/Matthias-Wandel/jhead.git && cd jhead && make CC="clang" -e CFLAGS="-g -fsanitize=address" -e LDFLAGS="-g -fsanitize=address"

Steps to Reproduce:
1.just run the following command

```
./jhead -purejpg ./tests_61786.jpg
```

poc:
tests_61786.zip

Actual results:

AddressSanitizer Report

```
$ ./jhead -purejpg ./tests_61786.jpg
=================================================================
==26249==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60e0000000e0 at pc 0x000000494f5f bp 0x7ffc20743730 sp 0x7ffc20742ef8
READ of size 160 at 0x60e0000000e0 thread T0
    #0 0x494f5e in __asan_memmove (/root/fuzz/jhead/jhead+0x494f5e)
    #1 0x4d14ff in RemoveSectionType /root/fuzz/jhead/jpgfile.c:669:13
    #2 0x4ca649 in ProcessFile /root/fuzz/jhead/jhead.c:1173:13
    #3 0x4c74e5 in main /root/fuzz/jhead/jhead.c:1759:13
    #4 0x7fc05b29283f in __libc_start_main /build/glibc-e6zv40/glibc-2.23/csu/../csu/libc-start.c:291
    #5 0x41b858 in _start (/root/fuzz/jhead/jhead+0x41b858)

0x60e0000000e0 is located 0 bytes to the right of 160-byte region [0x60e000000040,0x60e0000000e0)
allocated by thread T0 here:
    #0 0x4959c9 in realloc (/root/fuzz/jhead/jhead+0x4959c9)
    #1 0x4cf59f in CheckSectionsAllocated /root/fuzz/jhead/jpgfile.c:107:33
    #2 0x4ce3c0 in ReadJpegSections /root/fuzz/jhead/jpgfile.c:139:9
    #3 0x4cfd96 in ReadJpegFile /root/fuzz/jhead/jpgfile.c:381:11
    #4 0x4c8850 in ProcessFile /root/fuzz/jhead/jhead.c:908:10
    #5 0x4c74e5 in main /root/fuzz/jhead/jhead.c:1759:13
    #6 0x7fc05b29283f in __libc_start_main /build/glibc-e6zv40/glibc-2.23/csu/../csu/libc-start.c:291

SUMMARY: AddressSanitizer: heap-buffer-overflow (/root/fuzz/jhead/jhead+0x494f5e) in __asan_memmove
Shadow bytes around the buggy address:
  0x0c1c7fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c1c7fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c1c7fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c1c7fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c1c7fff8000: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
=>0x0c1c7fff8010: 00 00 00 00 00 00 00 00 00 00 00 00[fa]fa fa fa
  0x0c1c7fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c1c7fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c1c7fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c1c7fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c1c7fff8060: 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
==26249==ABORTING
```

Additional info:

Founder: giantbranch of NSFOCUS Security Team

Matthias-Wandel commented on Mar 4, 2021 Owner

Fixed by `a50953a`

Matthias-Wandel closed this as completed on Mar 4, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants