TALOS-2026-2330

LibRaw HuffTable::initval heap-based buffer overflow vulnerability

APRIL 7, 2026

CVE NUMBER

SUMMARY

CONFIRMED VULNERABLE VERSIONS

PRODUCT URLS

CVSSV3 SCORE

CWE

DETAILS

bits

HuffTable::initval

src/decompressors/losslessjpeg.cpp

```
void HuffTable::initval(uint32_t _bits[17], uint32_t _huffval[256], bool _dng_bug)
{
    memmove(bits, _bits, sizeof(bits));
    memmove(huffval, _huffval, sizeof(huffval));
    dng_bug = _dng_bug;

[1] nbits = 16;
    for(int i = 0; i < 16; i++)
    {
        if(bits[16 - i] != 0)
            break;
        nbits--;
    }
[2] hufftable.resize( size_t(1ULL << nbits));
    for (unsigned i = 0; i < hufftable.size(); i++) hufftable[i] = 0;

    int h = 0;
    int pos = 0;
    for (uint8_t len = 0; len < nbits; len++)
    {
[3]   for (uint32_t i = 0; i < bits[len + 1]; i++)
      {
        for (int j = 0; j < (1 << (nbits - len - 1)); j++)
        {
[4]       hufftable[h] = ((len+1) << 16) | (uint8_t(huffval[pos] & 0xff) << 8) |
uint8_t(shiftval[pos] & 0xff);
          h++;
        }
        pos++;
      }
    }
    [...]
}
```

_bits
bits bits
bits[0] bits[1] bits[16] bits[1] = 0
bits[2] = 2
bits[5] = 10

[1] nbits [2]
2^nbits [3]
len+1 2^(nbits-len-1) [4]

∑ bits[len+1] × 2^(nbits - len - 1) for len = 0 to nbits-1

bits
bits
[2] bits[5] = 100 bits[6] = 1
bits[i] = 0

nbits = 6 2^6 = 64 100 × 2^(6-5) + 1 × 2^(6-6) = 100 × 2 + 1 × 1 = 201 entries

[2] [4]

---

[2]

$$\sum 2^{(-L_i)} \leq 1$$

bits[i] bits[i]
2^(-i) ∑ bits[i] × 2^(-i) ≤ 1 for i = 1 to 16

nbits bits[i] != 0 i > nbits
bits[i] = 0 i = 1 to nbits ∑ bits[i] × 2^(-i) ≤ 1 for i = 1 to nbits

len = i - 1 ∑ bits[len + 1] × 2^(-len - 1) ≤ 1 for len = 0 to nbits-1

2^nbits ∑ bits[len + 1] × 2^(nbits - len - 1) ≤ 2^nbits for len = 0 to nbits-1

hufftable

Crash Information

```
=================================================================
==91815==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x606000000360 at
pc 0x00010099c5d8 bp 0x00016f479590 sp 0x00016f479588
WRITE of size 4 at 0x606000000360 thread T0
    #0 0x00010099c5d4 in HuffTable::initval(unsigned int*, unsigned int*, bool)
losslessjpeg.cpp:374
    #1 0x000100999a94 in LibRaw_LjpegDecompressor::initialize(bool, bool)
losslessjpeg.cpp:125
    #2 0x000100999f50 in LibRaw_LjpegDecompressor::LibRaw_LjpegDecompressor(unsigned
char*, unsigned int) losslessjpeg.cpp:48
    #3 0x0001009f17a4 in LibRaw::sony_ycbcr_load_raw() sonycc.cpp:276
    #4 0x000100a06b64 in LibRaw::unpack() unpack.cpp:447
    #5 0x0001008c8e4c in main poc_huffman.cpp:32
    #6 0x00018f671d50  (<unknown module>)

0x606000000360 is located 0 bytes after 64-byte region
[0x606000000320,0x606000000360)
allocated by thread T0 here:
    #0 0x000101223428 in _Znwm+0x74
(libclang_rt.asan_osx_dynamic.dylib:arm64e+0x4b428)
    #1 0x00010099ffcc in std::__1::vector<unsigned int, std::__1::allocator<unsigned
int>>::__append(unsigned long) vector.h:943
    #2 0x00010099c064 in HuffTable::initval(unsigned int*, unsigned int*, bool)
losslessjpeg.cpp:363
    #3 0x000100999a94 in LibRaw_LjpegDecompressor::initialize(bool, bool)
losslessjpeg.cpp:125
    #4 0x000100999f50 in LibRaw_LjpegDecompressor::LibRaw_LjpegDecompressor(unsigned
char*, unsigned int) losslessjpeg.cpp:48
    #5 0x0001009f17a4 in LibRaw::sony_ycbcr_load_raw() sonycc.cpp:276
    #6 0x000100a06b64 in LibRaw::unpack() unpack.cpp:447
    #7 0x0001008c8e4c in main poc_huffman.cpp:32
    #8 0x00018f671d50  (<unknown module>)

SUMMARY: AddressSanitizer: heap-buffer-overflow losslessjpeg.cpp:374 in
HuffTable::initval(unsigned int*, unsigned int*, bool)
Shadow bytes around the buggy address:
  0x606000000080: 00 00 00 00 00 00 00 00 fa fa fa fa 00 00 00 00
  0x606000000100: 00 00 00 00 fa fa fa fa 00 00 00 00 00 00 02 fa
  0x606000000180: fa fa fa fa 00 00 00 00 00 00 00 00 fa fa fa fa
  0x606000000200: 00 00 00 00 00 00 04 fa fa fa fa fa 00 00 00 00
  0x606000000280: 00 00 06 fa fa fa fa fa fd fd fd fd fd fd fd fd
=>0x606000000300: fa fa fa fa 00 00 00 00 00 00 00 00[fa]fa fa fa
  0x606000000380: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x606000000400: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x606000000480: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
```

```
  0x606000000500: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x606000000580: 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
==91815==ABORTING
```

TIMELINE

CREDIT

VULNERABILITY REPORTS

NEXT REPORT

TALOS-2026-2331

PREVIOUS REPORT

TALOS-2026-2365