COMMUNITY CONTENT

Home › Submit › 780538

Submit #780538: LibRaw 0.22.0 Out-of-bounds Write

Title LibRaw 0.22.0 Out-of-bounds Write

Description A heap out-of-bounds write exists in HuffTable::initval() (src/decompressors/losslessjpeg.cpp). The function builds a Huffman lookup table from JPEG DHT segment data, sizing the table to 1 << nbits entries. However, the number of entries actually written is derived from attacker-controlled bits[] values with no validation that the total write count fits within the allocation. A DHT segment with bits[1]=3 (three 1-bit codes) causes three writes into a 2-entry table, writing one entry past the end of the std::vector.

Technical Details:

Vulnerability Type: oob_write
File: decompressors/losslessjpeg.cpp
Affected Code:

```
350: void HuffTable::initval(uint32_t _bits[17], uint32_t _huffval[256], bool _dng_bug)
351: {
352:     memmove(bits, _bits, sizeof(bits));
353:     memmove(huffval, _huffval, sizeof(huffval));
354:     dng_bug = _dng_bug;
...
363:     hufftable.resize( size_t(1ULL << nbits));    // allocated size = 2^nbits
364:     for (unsigned i = 0; i < hufftable.size(); i++) hufftable[i] = 0;
365:
366:     int h = 0;
367:     int pos = 0;
368:     for (uint8_t len = 0; len < nbits; len++)
369:     {
370:       for (uint32_t i = 0; i < bits[len + 1]; i++)  // <-- bits[] from attacker-controlled
371:       {
372:         for (int j = 0; j < (1 << (nbits - len - 1)); j++)
373:         {
374:           hufftable[h] = ((len+1) << 16) | (uint8_t(huffval[pos] & 0xff) << 8) |
uint8_t(shiftval[pos] & 0xff);          // <-- no bounds check on h
375:           h++;
376:         }
377:         pos++;
378:       }
379:     }
```

Root Cause:

Community Content

Submissions are made by VulDB community users. VulDB is *not responsible* for their content nor the links to external sources.

Please use the raw information shown and the links listed *with caution*. They might contain malicious and harmful actions, code or data.

The corresponding VulDB entries contain the moderated, verified, and normalized information provided within the raw submission.

Documentation

- Submission Policy
- Data Processing
- CVE Handling

The table is sized to 1 << nbits. The total number of writes is:

Σ bits[len+1] × 2^(nbits-len-1) for len in [0, nbits-1]

For a valid canonical Huffman tree this sum equals exactly 1 << nbits, but nothing enforces this. The DHT parser (parse_dht()) only checks that the total symbol count is ≤ 256 — it does not verify code-space validity.

PoC:
A TIFF file wrapping a minimal lossless JPEG tile with a malformed DHT:
poc_ljpeg_hufftable.tif (https://github.com/binitamf/pocs/tree/main/libraw_ljpeg)

Build command used:

```
make -f Makefile.dist \
  CFLAGS="-O1 -I. -w -DUSE_ZLIB -fsanitize=address,undefined -fno-omit-frame-pointer" \
  LDADD="-lz -fsanitize=address,undefined"
```

Note: we use simple_dcraw in this poc but most of the shipped binaries also demonstrate the bug.

```
./LibRaw/bin/simple_dcraw poc_ljpeg_hufftable.tif
```

Sanitizer Output:

```
==PID==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x... at pc 0x...
WRITE of size 4 at 0x... thread T0
    #0  in HuffTable::initval(unsigned int*, unsigned int*, bool)
        decompressors/losslessjpeg.cpp:374
    #1  in LibRaw_LjpegDecompressor::initialize(bool, bool)
    #2  in LibRaw_LjpegDecompressor::LibRaw_LjpegDecompressor(unsigned char*, unsigned int)
    #3  in LibRaw::sony_ycbcr_load_raw()
    #4  in LibRaw::unpack()
    #5  in main

0x... is located 0 bytes after 8-byte region [0x...,0x...)
allocated by thread T0 here:
    #0  in operator new(unsigned long)
    #1  in std::vector<unsigned int>::_M_default_append(unsigned long)
    #2  in HuffTable::initval(...)

SUMMARY: AddressSanitizer: heap-buffer-overflow in HuffTable::initval(unsigned int*, unsigned int*, bool)
```

ref: https://github.com/LibRaw/LibRaw/issues/794

Source	https://github.com/binitamf/pocs/tree/main/libraw_ljpeg
User	binitam (UID 94731)
Submission	03/15/2026 10:34 PM (23 days ago)
Moderation	04/01/2026 02:43 PM (17 days later)
Status	Confirmed

VulDB entry [LibRaw up to 0.22.0 JPEG DHT Parser losslessjpeg.cpp HuffTable::initval bits[] out-of-bounds write]

Points 20