Integer overflow in DWA decoder outBufferEnd pointer arithmetic (missed variant of CVE-2026-34589)

High **cary-ilm** published **GHSA-m5qw-23x2-6phj** 2 days ago

Package

OpenEXR

Affected versions

3.2.0–3.2.7, 3.3.0–3.3.9, 3.4.0–3.4.9

Patched versions

3.2.8, 3.3.10, 3.4.10

Description

Summary

`internal_dwa_compressor.h:1040` performs `chan->width * chan->bytes_per_element` in `int32` arithmetic without a `(size_t)` cast. This is the same overflow pattern fixed in other decoders by CVE-2026-34589/34588/34544, but this line was missed.

Details

In `DwaCompressor_uncompress()`, line 1040:

```
outBufferEnd += chan->width * chan->bytes_per_element;
```

`chan->width` is `int32_t` (from the EXR dataWindow header), `chan->bytes_per_element` is `int8_t` (4 for FLOAT). The multiplication is done in signed 32-bit. For `width > 536,870,912` with a FLOAT channel, the product exceeds `INT32_MAX` and wraps negative.

`outBufferEnd` is `uint8_t*` — the wrapped value causes it to advance by a wrong amount. `DctCoderChannelData_push_row` then stores row pointers that alias earlier buffer regions. When `LossyDctDecoder_execute` writes decoded pixels through these pointers, it corrupts the heap.

The same file already has the correct pattern on other lines:

- Line 1215: `(size_t) chan->width * (size_t) pixelSize` — cast present
- Line 1699: `(size_t) curc->width * (size_t) curc->height` — cast present
- Line 1040: no cast — missed

The file `internal_dwa_compressor.h` has no git changes related to integer overflow fixes. The recent patches (commits `7c31424`, `088859f`, `e464a33`, `f5beec2`, `cf9bf84`) addressed PIZ, B44, PXR24, `generic_unpack()`, and `LossyDctDecoder_execute`, but not this code path.

Reachability

```
Crafted EXR (DWAA/DWAB compression, dataWindow width > 537M)
  → exr_start_read() parses header (default max image size = 0, no limit)
  → exr_decoding_run()
  → internal_exr_undo_dwa() → DwaCompressor_uncompress()
  → line 1040: int32 overflow in pointer arithmetic
  → aliased row pointers → heap corruption in LossyDctDecoder_execute
```

No default image size limit is enforced (`sMaxW = 0` in `base.c:128`).

Suggested fix

```
outBufferEnd += (size_t) chan->width * (size_t) chan->bytes_per_element;
```

Same pattern already used on lines 1215 and 1699 in the same file.

Impact

Heap buffer corruption when opening a crafted DWAA/DWAB EXR file. Same impact class as CVE-2026-34589 — OOB write via integer overflow in DWA decoder.

Severity

High 8.4 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Local
Attack Complexity	Low

Attack Requirements	None
Privileges Required	None
User interaction	Active
Vulnerable System Impact Metrics	
Confidentiality	High
Integrity	High
Availability	High
Subsequent System Impact Metrics	
Confidentiality	None
Integrity	None
Availability	None

Learn more about base metrics

CVSS:4.0/AV:L/AC:L/AT:N/PR:N/UI:A/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

CVE ID

CVE-2026-40250

Weaknesses

▶ CWE-190

Credits

 **Medoedus** Reporter