

SexyShoelessGodofWar / LibTiff-4.7.0-Write-What-Where Public[Code](#) [Issues 1](#) [Pull requests](#) [Actions](#) [Projects](#) [Security and quality](#)

1 Branch



0 Tags



Go to file

Go to file

<> Code



SexyShoelessGodofWar

Update README.md

9e4e78b · 7 months ago



README.md

Update README.md

7 months ago

README

LibTiff-4.7.0-Write-What-Where - CVE-2025-9900

This is a placeholder for a vulnerability discovered in LibTiff

Vulnerability Summary

Write-What-Where in libtiff via TIFFReadRGBAImageOriented

The vulnerability resides in the raster decoding logic of libtiff, specifically when processing paletted (indexed color) images with malformed metadata. The function `TIFFReadRGBAImageOriented()` computes a pointer offset into the raster buffer based on user-controlled image metadata:

```
raster + (rheight - img.height) * rwidth
```

If the attacker supplies a very large value for `img.height` (e.g., `0xFFFF`) and a valid `rheight` (e.g., `256`), this computation results in a large positive offset, causing the raster pointer (`cp`) passed into functions like `put8bitcmaptile()` or `put1bitbwtile()` to point beyond the bounds of the allocated buffer.

Inside those functions, memory writes occur like this:

```
*cp++ = PALmap[*pp][0];
```

- The write address (cp) is attacker-controlled via the offset calculation from `img.height`.
- The value written (`PALmap[*pp][0]`) is also attacker-controlled:
 - `*pp` is dereferenced from pixel data in the image file.
 - `PALmap` is constructed from the image's color palette, which the attacker also controls.



This constitutes a write-what-where vulnerability with a attacker control. Exploitation of a write-what-where primitive can lead to denial of service or code execution through supply of maliciously crafted files.

Version

4.7.0

Steps to reproduce

1. COmpile harness.c `clang -O0 -g -Ilibtiff -Ibuild-clean -o tiff_poc_crasher poc_crasher.c build-clean/libtiff/libtiff.a -lz -lzstd -lwebp -lwebpdemux -ldeflate -llzma -ljpeg -lm -ljpeg -lLerc`
2. `./tiff_fuzz_clean ./crashfile1.tiff`

This should create a seg fault.

The Code POC and tiff files are attached - as are the python files to generate a malicious one.

Note: *testGen.py* will generate a crash .tiff file. (i'm unable to upload the .tiff file)

I originally created this as a confidential issue - but didn't seem to have any eyes on it.

Platform

This was tested on: Distributor ID: Ubuntu Description: Ubuntu 24.04.2 LTS Release: 24.04 Codename: noble

NOTES

- This was reported, and fixed

References & Credits

Researcher - Gareth C

Company - AnchorSec Ltd. (<https://www.anchorsec.co.uk>)

Blog - <https://www.anchorsec.co.uk/insights/having-a-tiff-a-complicated-love-affair-with-vulnerability-research>

CVE - <https://access.redhat.com/security/cve/cve-2025-9900>

Releases

No releases published

Packages

No packages published

Contributors 1



SexyShoelessGodofWar