

# Commit c1b0318



ctruta committed 2 weeks ago · ✓ 4 / 4

fix: Sync info\_ptr->palette after in-place transforms

Copy `png\_ptr->palette` into `info\_ptr->palette` upon entering the function that runs immediately after the in-place transforms.

The palette decoupling in the previous commit gave `png\_struct` and `png\_info` independently-allocated palette buffers, fixing a use-after-free vulnerability. However, `png\_init\_read\_transformations` modifies `png\_ptr->palette` in place (e.g. for gamma correction or background compositing), and the old aliasing made those modifications visible through `png\_get\_PLTE`. With independent buffers, `info\_ptr->palette` retained the original values, causing our tests to fail on indexed-colour background compositing.

libpng18 · v1.6.56

1 parent [7ea9eea](#) commit c1b0318

1 file changed +15 -0 lines changed

Top



pngtran.c

1 file changed +15 -0 lines changed



▼ pngtran.c

```

@@ -2070,6 +2070,21 @@ png_read_transform_info(png_structrp png_ptr,
png_inforp info_ptr)
2070 2070 {
2071 2071     png_debug(1, "in png_read_transform_info");
2072 2072
2073 +     if (png_ptr->transformations != 0)
2074 +     {

```

```
2075 +     if (info_ptr->color_type == PNG_COLOR_TYPE_PALETTE &&
2076 +         info_ptr->palette != NULL && png_ptr->palette != NULL)
2077 +     {
2078 +         /* Sync info_ptr->palette with png_ptr->palette.
2079 +          * The function png_init_read_transformations may have modified
2080 +          * png_ptr->palette in place (e.g. for gamma correction or for
2081 +          * background compositing).
2082 +          */
2083 +         memcpy(info_ptr->palette, png_ptr->palette,
2084 +             PNG_MAX_PALETTE_LENGTH * (sizeof (png_color)));
2085 +     }
2086 + }
```

```
2073 2088 #ifndef PNG_READ_EXPAND_SUPPORTED
2074 2089     if ((png_ptr->transformations & PNG_EXPAND) != 0)
2075 2090     {
```



## Comments 0



Please [sign in](#) to comment.