November 17, 2020

GHSL-2020-142: Heap memory corruption in png-img - CVE-2020-28248

Bas Alberts

Summary

The png-img package provides NAN bindings for libpng. These bindings are vulnerable to an integer overflow which results in an underallocation of heap memory and subsequent heap memory corruption.

Additionally, the png-img package currently ships with libpng version 1.6.14 which contains several known and potentially exploitable vulnerabilities. The current public release version of libpng is 1.6.37.

Product

png-img

Tested Version

2.3.0

Details

Issue 1: Integer overflow resulting in heap overflow in `PngImg.cc:PngImg::InitStorage_()`

When loading a PNG image for processing by libpng the png-img bindings employ the `PngImg::InitStorage` function to allocate the initial memory required for the user supplied PNG data.

```
void PngImg::InitStorage_() {
    rowPtrs_.resize(info_.height, nullptr);
[1]
    data_ = new png_byte[info_.height * info_.rowbytes];

[2]
    for(size_t i = 0; i < info_.height; ++i) {
        rowPtrs_[i] = data_ + i * info_.rowbytes;
    }
}
```

At [1] we observe the allocation of a `png_byte` array of size `info_.height * info_.rowbytes`. Both the height and rowbytes structure members are of type `png_uint_32`, which implies the integer arithmetic expression here is explicitly an unsigned 32bit integer operation.

`info_.height` may be directly supplied from a PNG file as a 32bit integer and `info_.rowbytes` is derived from the PNG data as well.

This multiplication may trigger an integer overwrap which results in an underallocation of the `data_` memory region.

For example, if we set `info_.height` to 0x01000001 with an `info_.rowbytes` value of 0x100, the resulting expression would be (0x01000001 * 0x100) & 0xffffffff == 0x100, and as a result `_data` would underallocated as a 0x100 sized png_byte array.

As a consequence, at [2], the `rowPtrs_` array will be populated with row-data pointers that point outside of the bounds of the allocated memory region since the for loop conditional operates on the original `info_.height` value.

Subsequently, once the actual row data is read from the PNG file, any adjacent memory to the `data_` region may be overwritten with attacker controlled row-data up to `info_.height * info_.rowbytes` bytes, affording a great deal of process memory control to any would-be attacker.

Note that this overwrite may be halted early according to attacker wishes by simply not supplying sufficient amounts of row-data from the PNG itself, at which point libpng error routines would kick in. Any subsequent program logic handling the error paths would then operate on corrupted heap memory. This results in a highly controlled heap overflow in terms of both contents and size.

Impact

This issue may lead to arbitrary code execution if png-img is used to operate on untrusted PNG files.

Issue 2: vulnerable version of libpng shipped with png-img

The png-img package currently ships with libpng version 1.6.14 which contains several known and potentially exploitable vulnerabilities. The current public release version of libpng is 1.6.37.

It is recommended to update to the latest release version of libpng to prevent future abuse of known libpng vulnerabilities exposed via the png-img package.

CVE

- CVE-2020-28248

Coordinated Disclosure Timeline

- 08/05/2020: Maintainer notified
- 08/06/2020: Maintainer acknowledges report
- 10/05/2020: Maintainer prepares PR that fixes integer overflow and upgrades libpng version
- 11/03/2020: Maintainer releases version 3.1.0 to npm.

Resources

- https://github.com/github/securitylab-vulnerabilities/blob/main/vendor_reports/resources/GHSL-2020-142/x.png

Credit

This issue was discovered and reported by GHSL team member @anticomputer (Bas Alberts).

Contact

You can contact the GHSL team at `securitylab@github.com`, please include a reference to `GHSL-2020-142` in any communication regarding this issue.