New issue

Jump to bottom

# (CVE-2020-25219) pac server can trigger unbounded recursion in url.cpp recvline() #134

Closed **mcatanzaro** opened this issue on Sep 7, 2020 · 1 comment · Fixed by #136

**mcatanzaro** commented on Sep 7, 2020 Contributor

I found this in url.cpp:

```
static inline string recvline(int fd) {
        // Read a character.
        // If we don't get a character, return empty string.
        // If we are at the end of the line, return empty string.
        char c = '\0';

        if (recv(fd, &c, 1, 0) != 1 || c == '\n')
                return "";

        return string(1, c) + recvline(fd);
}
```

Looks like the server that hosts the proxy authconfig file can cause libproxy to overflow the stack by sending an unending stream of characters without a newline. The PAC server should be trusted to not do that, but it's still not good. Normal use with a non-malicious server looks like this:

```
(gdb) bt
#0  recvline (fd=4) at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:389
#1  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#2  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#3  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#4  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#5  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#6  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#7  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#8  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#9  0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#10 0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#11 0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#12 0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#13 0x00007ffff7f987ac in recvline (fd=<optimized out>)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:398
#14 0x00007ffff7f99749 in libproxy::url::get_pac (this=this@entry=0x7fffffffdd30)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/url.cpp:464
#15 0x00007ffff7f8b7f9 in libproxy::proxy_factory::expand_pac (this=0x416eb0, confurl=...)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/proxy.cpp:393
#16 0x00007ffff7f9120a in libproxy::proxy_factory::get_proxies (this=0x416eb0, realurl=...)
    at /usr/src/debug/libproxy-0.4.15-17.fc32.x86_64/libproxy/proxy.cpp:215
#17 0x00007ffff7f916dc in px_proxy_factory_get_proxies (self=0x416eb0, url=0x402010 "https://lwn.net")
    at /usr/include/c++/10/bits/char_traits.h:300
#18 0x0000000000401188 in main ()
```

**mcatanzaro** mentioned this issue on Sep 9, 2020

**Rewrite url::recvline to be nonrecursive** #136

Merged

**mcatanzaro** changed the title ~~pac server can trigger unbounded recursion in url.cpp recvline()~~ (CVE-2020-25219) pac server can trigger unbounded recursion in url.cpp recvline() on Sep 9, 2020

**mcatanzaro** commented on Sep 9, 2020 Contributor Author

We received CVE-2020-25219 for this issue.

👍 1

**DimStar77** closed this as completed in #136 on Sep 10, 2020

**DimStar77** added a commit that referenced this issue on Sep 10, 2020

`Merge pull request #136 from mcatanzaro/mcatanzaro/#134` ··· `836c10b`

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

**Rewrite url::recvline to be nonrecursive**

1 participant