


[Products](#)
[Services](#)
[Publications](#)
[Resources](#)
[What's new](#)

Follow @Openwall on Twitter for new release announcements and other news

[<prev] [next>] [<thread-prev] [thread-next>] [day] [month] [year] [list]

Message-ID: <CAFRnB2X3U2K6e14XLpJo7X6bWbRD6mSGsfzmWzReJbQnDPvEhQ@mail.gmail.com>
 Date: Wed, 2 Nov 2022 13:26:37 -0400
 From: Alex Gaynor <alex.gaynor@...il.com>
 To: oss-security@...ts.openwall.com
 Subject: Re: Re: OpenSSL X.509 Email Address 4-byte Buffer Overflow (CVE-2022-3602), X.509 Email Address Variable Length Buffer Overflow (CVE-2022-3786)

The distinction I'd make is that Rust's behavior is guaranteed, while the factors in C leading to a buffer overflow being unexploitable are contingent. Users compiling without -fstack-protector-strong, precise allocation patterns or stack layout patterns, etc all impact whether a C buffer overflow is exploitable or not.

It's telling that OpenSSL originally understood this to be a CRITICAL severity, but only after analysis and feedback from many other folks were they confident enough to lower it a HIGH severity -- in Rust one would know right off that bat that it was definitely a DoS at worst.

And of course, many buffer overflows never get the deep expert analysis required to establish if they're exploitable or not -- I don't need to tell you that the P0 blog is full of exploits of 1-byte buffer overflows that many people wrote off as "no way that can be exploited" :-)

Alex

On Wed, Nov 2, 2022 at 1:19 PM Tavis Ormandy <tavis@...il.com> wrote:

```
>
> On 2022-11-02, Alex Gaynor wrote:
> > In Rust, assuming you wrote normal safe Rust[0], and you had code that
> > overran a buffer on the stack, you'd get a panic() -- which is roughly
> > an abort (there's even a mode where it literally is an abort. By
> > default it unwinds and runs destructors and such). As a general rule,
> > bounds check issues aren't caught at compile time (in contrast with
> > temporal safety, which mostly is enforced at compile time.)
> >
> >
> Got it - thanks! It seems like in the specific case of non-exploitable
> overflows, rust wouldn't have made too much difference (abort() vs
> panic())... although obviously that doesn't mean other issues wouldn't
> have been mitigated.
```

> Tavis.

```
> --
> _o) $ lynx lock.cmpxchg8b.com
> _\ \ _o) _o) $ finger tavis@...org
> _\_V _(\ ) _(\ ) @tavis
>
```

--
 All that is necessary for evil to succeed is for good people to do nothing.

[Powered by blists - more mailing lists](#)

Please check out the [Open Source Software Security Wiki](#), which is counterpart to this [mailing list](#).

Confused about [mailing lists](#) and their use? [Read about mailing lists on Wikipedia](#) and check out these [guidelines on proper formatting of your messages](#).