hyperium / **hyper** Public

Code | Issues 173 | Pull requests 25 | Discussions | Actions | Security 3 | ...

Lenient Parsing of Content-Length Header When Prefixed with Plus Sign

Low **seanmonstar** published **GHSA-f3pg-qwvg-p99c** on Jul 7, 2021

Package

hyper (crates.io)

Affected versions	Patched versions
< 0.14.10	0.14.10

Description

Summary

hyper's HTTP/1 server code had a flaw that incorrectly parses and accepts requests with a `Content-Length` header with a prefixed plus sign, when it should have been rejected as illegal. This combined with an upstream HTTP proxy that doesn't parse such `Content-Length` headers, but forwards them, can result in "request smuggling" or "desync attacks".

Vulnerability

The flaw exists in all prior versions of hyper, if built with `rustc` v1.5.0 or newer.

Example:

```
GET / HTTP/1.1
Host: example.com
Content-Length: +3

abc
```

This request gets accepted and hyper reads the body as abc. The request *should* be rejected, according to RFC 7230, since the ABNF for `Content-Length` only allows for `DIGIT` s. This is due to using the `FromStr` implementation for `u64` in the standard library. By differing from the spec, it is possible to send requests like these to endpoints that have different HTTP implementations, with different interpretations of the payload semantics, and cause "desync attacks".

In this particular case, an upstream proxy would need to error when parsing the `Content-Length` , but *not* reject the request (swallowing its own error), and forwarding the request as-is with the `Content-Length` still included. *Then* the upstream proxy and hyper would disagree on the length of the request body. The combination of these factors would be extremely rare.

Read more about desync attacks: https://portswigger.net/research/http-desync-attacks-request-smuggling-reborn

Impact

To determine if vulnerable, all these things must be true:

- **Using hyper as an HTTP server**. While the lenient decoder also exists in the client, a vulnerability does not exist around *responses*.
- **Using HTTP/1**. The HTTP/2 code uses a stricter parser.
- **Using a vulnerable HTTP proxy upstream to hyper**. If an upstream proxy correctly rejects the illegal `Content-Length` header, *OR* can parse the length with the plus sign, the desync attack cannot succeed.

Patches

We have released the following patch versions:

- v0.14.10 (to be released when this advisor is published)

Workarounds

Besides upgrading hyper, you can take the following options:

- Reject requests manually that contain a plus sign prefix in the `Content-Length` header.
- Ensure any upstream proxy handles `Content-Length` headers with a plus sign prefix.

Credits

This issue was initially reported by Mattias Grenfeldt and Asta Olofsson.

Severity

Low

CVE ID

CVE-2021-32715

Weaknesses

CWE-444

Credits

mattiasgrenfeldt

asta12