2

1-click DOS in fastify-static via directly passing user's input to new URL() of NodeJS without try/catch

Share:

TIMELINE

drstrnegth submitted a report to Fastify. Oct 6th (about 1 year ago)

Summary:

When fastify-static is mounted at root and registered the option `{ redirect: true }` (default of redirect option is `false`), the following line directly feed user's input which is `req.raw.url` to URL API without try/catch: https://github.com/fastify/fastify-static/blob/master/index.js#L439. A remote attacker can send a GET request to server with path = `//^/..` , this will cause the URL API to throw error and eventually crash the server.

Steps To Reproduce:

1. Download `fastify-dos.zip`
2. bash run.sh
3. Open your terminal and run: `curl --path-as-is "http://localhost:3000//^/.."`

After that the server will crash and return error `TypeError [ERR_INVALID_URL]: Invalid URL: //^/..` .

Fix proposal

You can add a try/catch to prevent crash. However, if you only fix by adding try/catch, attacker can still cause open redirect.

1. Run the server in my `fastify-dos.zip` again
2. Use Google Chrome and navigate to `http://localhost:3000//a//youtube.com/%2e%2e%2f%2e%2e` (I tested on Chrome, Firefox, Safari, Opera, Edge, worked on all of them)
3. You will see that you get redirected to `https://www.youtube.com/..%2F..`

I like the idea of fixing open redirect by having a base URL = `http://localhost.com/` as second parameter in https://github.com/fastify/fastify-static/blob/master/index.js#L439. However, I looked up on MDN spec about the URL API and I got surprised when I saw the last example at: https://developer.mozilla.org/en-US/docs/Web/API/URL/URL#examples, which is `new URL("//foo.com", "https://example.com") // => 'https://foo.com' (see relative URLs)` , this is the main reason why the open redirect bug is still persist.

To fix this bug, I think we can check leading slash of `req.raw.url` , and allow at most 1 leading slash `/` before attempt to redirect.

Impact

- Denial of service
- Open redirect

1 attachment:
F1473123: fastify-dos.zip

drstrnegth posted a comment. Updated Oct 7th (about 1 year ago)

Note: to DOS the server, please change the command at step 3 in ***Steps To Reproduce*** to the following command: `curl --path-as-is "http://localhost:3000//:/.."` .

The reason is it depends on the NodeJS version of the system, the payload `curl --path-as-is "http://localhost:3000//:/.."` trigger DOS in both version NodeJS v14 and v15, while the command `curl --path-as-is "http://localhost:3000//^/.."` only trigger DOS for NodeJS v15.

eomm joined this report as a participant. Oct 9th (about 1 year ago)

eomm posted a comment. Updated Oct 9th (about 1 year ago)

Hi,
do you think this fix cover all the cases? (see the diff attached)

Thanks for your support

1 attachment:
F1476429: From_532b4704e0b23b917d99b3584abc21f61aa.txt

drstrnegth posted a comment. Oct 9th (about 1 year ago)

I think there is 1 small case left:
1: run the server that has applied your fix
2: `curl -ik http://localhost:3000/\\a//google.com/%2e%2e%2f%2e%2e` . You will see the header: `location: //google.com/%2e%2e%2f%2e%2e/`

This is because the backslash '\' character is treated like forward slash '/' in the API `new URL()` . This redirect would not work in browser because browser will change the backslash '\' to forward slash '/' before sending request to fastify server. However, it still causes open redirect when other server send http request to fastify-static server.

To fix this ultimately, you can simply add 1 final condition:

Code 85 Bytes Wrap lines Copy Download

```
-   if (url.startsWith('//')) {
+  if (url.startsWith('//') || url.startsWith('/\\')) {
```

I think it will be fixed after this

eomm posted a comment. Oct 10th (about 1 year ago) mcollina Fastify staff closed the report and changed the status to ● Resolved. Oct 11th (about 1 year ago)

Fixed in v4.4.1

mcollina Fastify staff requested to disclose this report. Oct 11th (about 1 year ago)

eomm posted a comment. Oct 11th (about 1 year ago)

Hi

@drstrnegth we have a new open issue as regression: https://github.com/fastify/fastify-static/issues/246

It says that the double slashes are managed by popular proxies and the specs allow it but I did not found any evidence: https://datatracker.ietf.org/doc/html/rfc3986#section-3.3

Do you think that replacing the double (or more) slashes with one slash could be good from the security point of view?

Thanks again for your help

drstrnegth posted a comment. Oct 11th (about 1 year ago)

Hi, I think replacing the double (or more) slashes is good in security point of view. In fact, replacing double or more slash with 1 slash is how ExpressJS fixed this bug, here is the fixed commit: https://github.com/expressjs/serve-static/commit/0399e399935bab99530d6926094b4451438c2d50. Make sure you check both the forward slash '/' and the black slash '\'.

mcollina Fastify staff posted a comment. Oct 11th (about 1 year ago)

@drstrnegth could you please review https://github.com/fastify/fastify-static/pull/247?

climba03003 Fastify staff posted a comment. Updated Oct 11th (about 1 year ago)

Express solution is cleaner than my beginning use of `while` and `startsWith`. It use a counter and reduce a lot of comparison.
I updated the beginning part to the same as `Express` solution.

Checking the leading slash of path is good enough, as the later part of double and triple slash will not affect the redirection result of browser.

drstrnegth posted a comment. Oct 11th (about 1 year ago)

Hi @climba03003, I checkout to your commit at "e358026c1041d6686188a0865b2a740a50bf6e0b" and saw you commented:

Code 110 Bytes Wrap lines Copy Download

```
// in here the url is already santitize once
// all the / and \ are equal to /
function getRedirectUrl (url) {
```

Then I checked using: `curl -ik http://localhost:3000/\\a//google.com/%2e%2e%2f%2e%2e` I see that url still contains character '\'. Can you recheck that if url is sanitized and "all the / and \ are equal to /"?

I ran curl and saw this: see in my attachment

1 attachment:
F1478249: screenshot_fastify.png

climba03003 Fastify staff posted a comment. Updated Oct 11th (about 1 year ago)

It should be either `light-my-request` or other part doing it. I have update to check both forward and backward slash now.

I think it handle both of the case now.

1 attachment:
F1478268: Screenshot_2021-10-12_001515.png

drstrnegth posted a comment. Oct 11th (about 1 year ago)

Okay I see your fix is good now.

drstrnegth agreed to disclose this report. Oct 11th (about 1 year ago)

This report has been disclosed. Oct 11th (about 1 year ago)