New issue

Server-Side Request Forgery Vulnerability (CVE-2024-39338) #6463

Closed #6539

jeffhacks opened on Jun 24, 2024

Describe the bug

Axios is vulnerable to a Server-Side Request Forgery attack caused by unexpected behaviour where requests for path relative URLS gets processed as protocol relative URLs.

This could be leveraged by an attacker to perform arbitrary requests from the server, potentially accessing internal systems or exfiltrating sensitive data.

To Reproduce

In this vulnerable code snippet, the developer intended to invoke a path relative to the base URL, however it allows an attacker to craft a malicious protocol-relative URL which is then requested by the server. Given protocol-relative URLs are not relevant server-side as there is no protocol to be relative to, the expected result would be an error. Instead, a valid URL is produced and requested.

Example Vulnerable Code

```
const axios = require('axios');

this.axios = axios.create({
  baseURL: 'https://userapi.example.com',
});

//userId = '12345';
userId = '/google.com'

this.axios.get(`/${userId}`).then(function (response) {
  console.log(`config.baseURL:  ${response.config.baseURL}`);
  console.log(`config.method:   ${response.config.method}`);
  console.log(`config.url:      ${response.config.url}`);
  console.log(`res.responseUrl: ${response.request.res.responseUrl}`);
});
```

Expected Output (Prior to axios 1.3.2):

```
(node:10243) UnhandledPromiseRejectionWarning: TypeError [ERR_INVALID_URL]: Invalid URL:
```

Developer might also have expected:

```
config.baseURL:  https://userapi.example.com
config.method:   get
config.url:      https://userapi.example.com//www.google.com/
res.responseUrl: https://userapi.example.com//www.google.com/
```

Observed Output:

```
config.baseURL:  https://userapi.example.com
config.method:   get
config.url:      //google.com
res.responseUrl: http://www.google.com/
```

This behaviour is potentially unexpected and introduces the potential for attackers to request URLs on arbitrary hosts other than the host in the base URL.

The code related to parsing and preparing the URL for server-side requests, prior to version 1.3.2, only passed one argument to the Node.js URL class.

```
const fullPath = buildFullPath(config.baseURL, config.url);
const parsed = new URL(fullPath);
const protocol = parsed.protocol || supportedProtocols[0];
```

Version 1.3.2 introduced `http://localhost` as a base URL for relative paths (#5458)

```
const fullPath = buildFullPath(config.baseURL, config.url);
const parsed = new URL(fullPath, 'http://localhost');
const protocol = parsed.protocol || supportedProtocols[0];
```

As protocol-relative URLs are considered to be absolute, the config.baseURL value is ignored so protocol-relative URLs are passed to the URL class without a protocol. The node.js URL class will then prepend the protocol from `'http://localhost'` to the protocol-relative URL.

For example

```
> new URL('//google.com', 'https://example.org');
URL {
  href: 'https://google.com/',
  ...
}

> new URL('//google.com', 'file://example.org');
URL {
  href: 'file://google.com/',
```

```
    ...
}
```

Code snippet

No response

Expected behavior

An error could be raised when attempting to request protocol-relative URLs server-side as unlike in a client-side browser session, there is no established protocol to be relative to.

Axios Version

No response

Adapter Version

No response

Browser

No response

Browser Version

No response

Node.js Version

No response

OS

No response

Additional Library Versions

No response

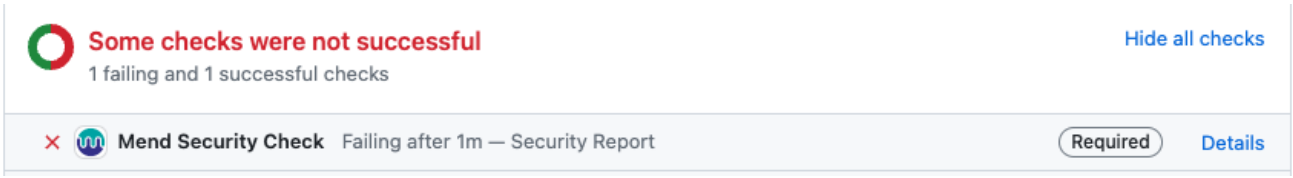
Additional context/Screenshots

No response

jeffhacks changed the title ~~Server-Side Request Forgery Vulnerability~~ Server-Side Request Forgery Vulnerability (CVE-2024-39338) on Jul 3, 2024

pinussilvestrus on Aug 12, 2024

Is there a fix plan for this? v1.7.3 is not fixing this and it's blocking releasing one of our libraries.



👍 47

SilverSting on Aug 12, 2024 · edited by SilverSting

Edits

Hi. Trying to understand this vulnerability better.

Mixed use of relative and absolute (including, protocol-relative) URLs

Axios will see `//google.com` as an argument to `get` call, since the text substitution will occur outside of the function call and that's an acceptable value, isn't it? It's quite possible that the same client instance could be used for fetching both relative URL and absolute URL based resources.

```
userId = '/google.com'
this.axios.get(`/${userId}`);
```

Of course this may not align with application developer's expectation (ie. to be relative path). But it feels like that's not Axios' problem to solve. Axios could potentially warn this use pattern (ie. use of baseUrl with non relative path), but it seems too harsh for axios to dictate this.

Impact of using protocol-relative URLs

The following code kinda forces `http` protocol if protocol-relative URLs are used.

```
const parsed = new URL(fullPath, 'http://localhost');
```

Is this the main concern here? I guess an attacker could potentially move away from non-http protocol (eg. local file location, deemed safe) to http protocol to deliver remotely located malicious files into the target system.

Expected behavior

> An error could be raised when attempting to request protocol-relative URLs server-side as unlike in a client-side browser session, there is no established protocol to be relative to.

If this is indeed a good solution, then `isAbsoluteURL` could be updated to ignore protocol-relative URLs to produce this output. Please note `config.url` is set by the `_request` implementation,, so it will be still `//google.com` . But requested url will be **combined** url.

```
config.baseURL:  https://userapi.example.com
config.method:   get
config.url:      //google.com
res.responseUrl: https://userapi.example.com//google.com
```

👍 5 👎 1 🚀 2

hainenber mentioned this on Aug 12, 2024

fix(sec): disregard protocol-relative URL to remediate SSRF #6539

hainenber on Aug 12, 2024 · edited by hainenber — Edits — Contributor

Guess most of us are here due to Snyk's action lately ;)
... Well, the enterprise-y us.

I made a draft PR to materialize **@SilverSting**'s idea above. Do lmk if that works out in the long-term, fellas

Edit: regression test added

👍 12 😄 1 👀 3

nathanloyer on Aug 13, 2024

Several auditing tools are reporting this issue as of today, including `npm audit`.

GHSA-8hc4-vh64-cxmj

👍 57

quldude mentioned this on Aug 13, 2024

Seeing a security vulnerability with the axios version used by this library. reportportal/client-javascript#210

torokati44 mentioned this on Aug 13, 2024

CVE-2024-39338 #6540

furkanmustafa on Aug 13, 2024 · edited by furkanmustafa — Edits

This vuln report seems like alerting.. all the alerts.. But I'm quite confused (maybe not enough caffeine today)

Question: "using unsanitized input/data as string in URLs" made towards elsewhere is insanely dangerous in any way? on any library any place and any location. I somehow cannot thing that axios is wrong here.

```
const unsanitizedVar = '/google.com'; // was expecting something like '12345'

const ANY_HTTP_LIBRARY = require(something, maybe axios);
const ANY_EXTERNAL_ENDPOINT = 'https://somewhere.innocent';

this.httpclient = ANY_HTTP_LIBRARY.create({
  baseURL: ANY_EXTERNAL_ENDPOINT,
});

// Isn't >> THIS -the imaginary use case below- << the problem?
this.httpclient.request({ url: `/${unsanitizedVar}` });
```

even if you can prevent this value from diverting request to another target, you cannot prevent it from doing other kind of malicious things with the assumptions of this vuln report (and the fix).. with my current understanding of it.

Axios's (current) baseURL behavior seems like a similar behavior of `src=` tags of a browser. Which is one possible, and (IMO) well defined design. (not saying "[in]correct"|"good"/"bad")

PS. (edit) Tested browser behavior, using `fetch`, when on "https://my.site";

```
> fetch('/12345').then(r => console.log('resp', r.url));
..
[XHR] GET https://my.site/12345

> fetch('//www.google.com').then(r => console.log('resp', r.url));
..
[XHR] GET https://www.google.com
```

Same Vuln report would apply to (the design of?) `fetch` of browsers?

19 👀 6

KentaHizume mentioned this on Aug 13, 2024

axiosに脆弱性(CVE-2024-39338) AlesInfiny/maris#1576

terozio mentioned this on Aug 13, 2024

NPM audit fails for anything depending on aws-crt and axios / CVE-2024-39338 aws/aws-sdk-js-v3#6381

sfc-gh-dszmolka mentioned this on Aug 13, 2024

SNOW-1622978: GHSA-8hc4-vh64-cxmj / CVE-2024-39338 / SNYK-JS-AXIOS-7361793 in axios >= 1.3.2 <= 1.7.3 snowflakedb/snowflake-connector-nodejs#887

marc2332 mentioned this on Aug 13, 2024

[Task (tooling/wallet)]: Update Axios once CVE-2024-39338 is patched iotaledger/iota#1763

rrkoshta123 on Aug 13, 2024

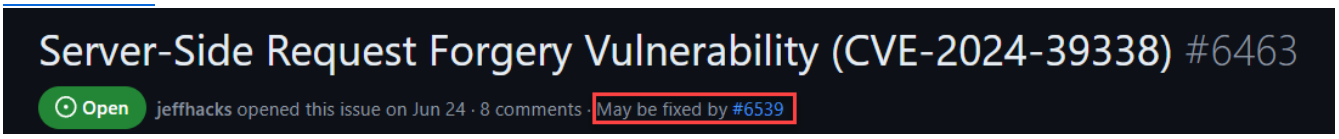
Is there any way that we can fix this issue, as it is showing no patch available.

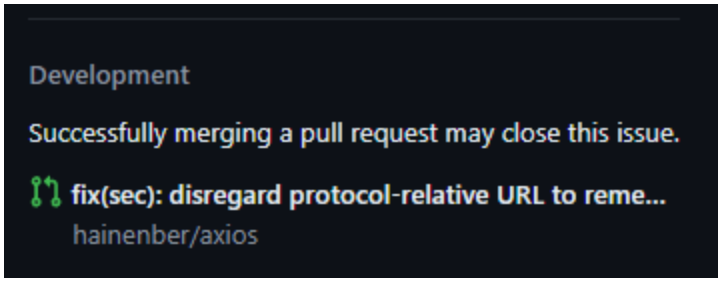
high	Server-Side Request Forgery in axios
Package	axios
Patched in	No patch available
Dependency of	axios
Path	axios
More info	https://www.npmjs.com/advisories/1098582

MikeMcC399 on Aug 13, 2024 · edited by MikeMcC399 Edits

@rrkoshta123

- See at the top of this issue, which points to fix(sec): disregard protocol-relative URL to remediate SSRF #6539





54 remaining items

Load more

renovate mentioned this in 12 pull requests 7 minutes ago

- fix(deps): update dependency axios to v1.8.2 [security] taevel02/sunday#30
- fix(deps): update dependency axios to v1.8.2 [security] b-sharpe-SA/pecari#92
- Update dependency axios to v1.8.2 [SECURITY] hmcts/labs-kainosrogery-nodejs#23
- fix(deps): update dependency axios to v1.8.2 [security] GiorgioBrux/nitro-sniper-enhanced#210
- Update dependency axios to v1.8.2 [SECURITY] fedora-infra/fmn#1299
- fix(deps): update dependency axios to v1.8.2 [security] Clivern/Cattle#717
- fix(deps): update dependency axios to v1.8.2 [security] SAP/open-ux-tools#3005
- Update dependency axios to v1.8.2 [SECURITY] TryGhost/vscode#62
- Update dependency axios to v1.8.2 [SECURITY] jhipster/jhipster-lite#12053
- build(deps): update dependency axios to v1.8.2 [security] wmfs/tymly-fastify-plugin#544
- fix(deps): update dependency axios to v1.8.2 [security] - autoclosed bytebase/bytebase#15455
- fix(deps): update dependency axios to v1.8.2 [security] yjl9903/Optc#540

renovate-bot mentioned this 5 minutes ago

- fix(deps): Update dependency axios to v1.8.2 [SECURITY] GoogleCloudPlatform/terraform-ml-image-annotation-gcf#211

openverse-bot mentioned this 3 minutes ago

- Update dependency axios to v1.8.2 [SECURITY] WordPress/openverse#5387

renovate mentioned this 1 minute ago

- chore(deps): update vulnerable [security] jellyfin/jellyfin-vue#2597

Assignees

No one assigned

Labels

No labels

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

fix(sec): disregard protocol-relative URL to remediate SSRF
axios/axios

Participants

 +12