`undici.request` vulnerable to SSRF using absolute URL on `pathname`

Moderate **mcollina** published **GHSA-8qr4-xgw6-wmr3** on Aug 9

Package

undici (npm)

Affected versions	Patched versions
=< 5.8.1	5.8.2

Description

Impact

`undici` is vulnerable to SSRF (Server-side Request Forgery) when an application takes in **user input** into the `path/pathname` option of `undici.request`.

If a user specifies a URL such as `http://127.0.0.1` or `//127.0.0.1`

```
const undici = require("undici")
undici.request({origin: "http://example.com", pathname: "//127.0.0.1"})
```

Instead of processing the request as `http://example.org//127.0.0.1` (or `http://example.org/http://127.0.0.1` when `http://127.0.0.1 is used`), it actually processes the request as `http://127.0.0.1/` and sends it to `http://127.0.0.1`.

If a developer passes in user input into `path` parameter of `undici.request`, it can result in an *SSRF* as they will assume that the hostname cannot change, when in actual fact it can change because the specified path parameter is combined with the base URL.

Patches

This issue was fixed in `undici@5.8.1`.

Workarounds

The best workaround is to validate user input before passing it to the `undici.request` call.

For more information

If you have any questions or comments about this advisory:

- Open an issue in undici repository
- To make a report, follow the SECURITY document

Severity

Moderate 5.3 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	Low
Integrity	None
Availability	None

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

CVE ID

CVE-2022-35949

Weaknesses

CWE-918

Credits

Haxatron