Path traversal in webpack-dev-middleware #23

Dismiss alert

Open Opened 3 weeks ago on **webpack-dev-middleware** (npm) · package-lock.json

Bump the npm_and_yarn group across 1 directory with 4 updates

Review security update

Merging this pull request would fix 1 Dependabot alert on webpack-dev-middleware in package-lock.json.

Package	Affected versions	Patched version
webpack-dev-... (npm)	<= 5.3.3	5.3.4

Summary

*The **webpack-dev-middleware** middleware does not validate the supplied URL address sufficiently before returning the local file. It is possible to access any file on the developer's machine.*

Details

The middleware can either work with the physical filesystem when reading the files or it can use a virtualized in-memory *memfs* filesystem.
If *writeToDisk* configuration option is set to **true**, the physical filesystem is used:
https://github.com/webpack/webpack-dev-middleware/blob/7ed24e0b9f53ad1562343f9f517f0f0ad2a70377/src/utils/setupOutputFileSystem.js#L21

Severity

High 7.4 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	Required
Scope	Changed
Confidentiality	High
Integrity	None
Availability	None

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

Tags

Development dependency
Patch available

Weaknesses

CWE-22

CVE ID

CVE-2024-29180

GHSA ID

GHSA-wr3j-pwj9-hqq6

See advisory in GitHub Advisory Database

See all of your affected repositories

The ***getFilenameFromUrl*** method is used to parse URL and build the local file path.
The public path prefix is stripped from the URL, and the **unsecaped** path suffix is appended to the *outputPath*:
https://github.com/webpack/webpack-dev-middleware/blob/7ed24e0b9f53ad1562343f9f517f0f0ad2a70377/src/utils/getFilenameFromUrl.js#L82
As the URL is not unescaped and normalized automatically before calling the midlleware, it is possible to use *%2e* and *%2f* sequences to perform path traversal attack.

PoC

*A blank project can be created containing the following configuration file **webpack.config.js**:*

```
module.exports = { devServer: { devMiddleware: {
writeToDisk: true } } };
```

When started, it is possible to access any local file, e.g. */etc/passwd*:

```
$ curl
localhost:8080/public/..%2f..%2f..%2f..%2f../etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
```

Impact

The developers using *webpack-dev-server* or *webpack-dev-middleware* are affected by the issue. When the project is started, an attacker might access any file on the developer's machine and exfiltrate the content (e.g. password, configuration files, private source code, ...).

If the development server is listening on a public IP address (or **0.0.0.0**), an attacker on the local network can access the local files without any interaction from the victim (direct connection to the port).

If the server allows access from third-party domains (CORS, ***Allow-Access-Origin: **), an attacker can send a malicious link to the victim. When visited, the client side script can connect to the local server and exfiltrate the local files.

Recommendation

The URL should be unescaped and normalized before any further processing.

 **dependabot** bot opened this 3 weeks ago