HIGH

Search by package name or CVE

Directory Traversal

Affecting fast-http package, versions *

INTRODUCED: 20 JUN 2020 CVE-2020-7687 CWE-22 FIRST ADDED BY SNYK

Share

How to fix?

A fix was pushed into the `master` branch but not yet published.

Overview

fast-http is a library that allows you to create a tiny web server.

Affected versions of this package are vulnerable to Directory Traversal. There is no path sanitization in the path provided at `fs.readFile` in `index.js`.

PoC by JHU System Security Lab

1. Start the server

```
var Server = require("fast-http"); var srv = new Server(8080,'./',true)
```

2. Create a file `sensitive-file` in the server's parent directory
3. `curl -s --path-as-is http://127.0.0.1:8080/../sensitive-file`

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 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 Commit
- GitHub PR

Snyk CVSS

Exploit Maturity	Proof of concept
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-FASTHTTP-572892
Published	20 Jun 2020
Disclosed	20 Jun 2020
Credit	JHU System Security Lab

Report a new vulnerability

Found a mistake?

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