

Directory Traversal

Affecting **bikeshed** package, versions [3.0.0)

INTRODUCED: 8 AUG 2021 CVE-2021-23423 CWE-22 FIRST ADDED BY SNYK Share

**How to fix?**

Upgrade bikeshed to version 3.0.0 or higher.

Overview

bikeshed is a pre-processor for spec documents.

Affected versions of this package are vulnerable to Directory Traversal. This can occur when an untrusted source file containing include , include-code or include-raw block is processed. The contents of arbitrary files could be disclosed in the HTML output.

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

PRODUCT

- Snyk Open Source
- Snyk Code
- Snyk Container
- Snyk Infrastructure as Code
- Test with Github
- Test with CLI

RESOURCES

- Vulnerability DB
- Documentation
- Disclosed Vulnerabilities

MEDIUM

Search by package name or CVE

Snyk CVSS

Attack Complexity	Low
User Interaction	Required
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-PYTHON-BIKESHED-1537647
Published	15 Aug 2021
Disclosed	8 Aug 2021
Credit	apple502j

Report a new vulnerability

Found a mistake?

[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](#)



© 2022 Snyk Limited

Registered in England and Wales. Company number: 09677925

Registered address: Highlands House, Basingstoke Road, Spencers Wood, Reading, Berkshire, RG7 1NT.