OWASP

NODE.JS SECURITY DONE RIGHT

Tips and Tricks They Won’t Teach You in School

Hewlett Packard Enterprise

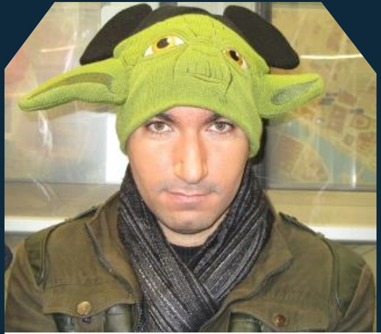
Liran Tal
R&D Team Lead for a Full-Stack Technology Web Marketplace

Hello!

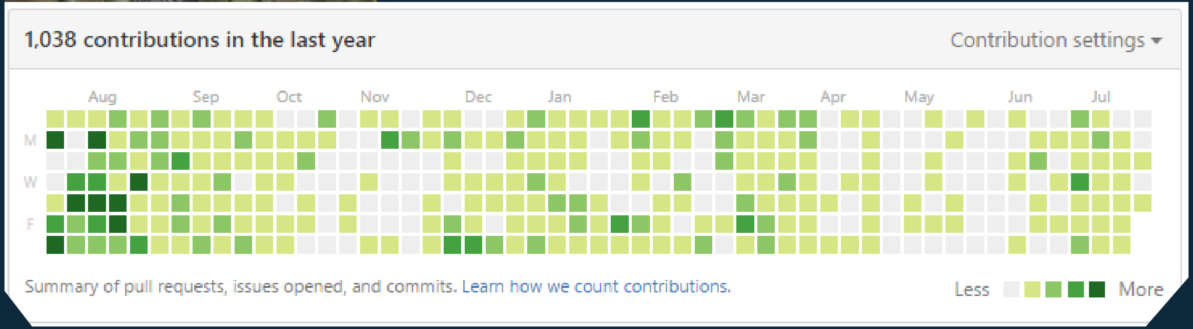
I am Liran Tal

R&D Team Lead for a Full-Stack Technology Web Marketplace

Hewlett Packard Enterprise



Open Source Evangelist



1,038 contributions in the last year

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Agenda

1. Node.js background
2. Security Horror Stories in Node.js
3. Tips & Recipes
   - Security by HTTP Headers
   - Secure Session Cookies
   - NoSQL Injection
   - Regular Expressions DOS attack
   - Dependencies Vuln' Scanning
   - Security as a Service

The Big Bang of JavaScript

Node.js Born in **2009**

- Open Source
- Cross-Platform
- Asynchronous JavaScript Runtime

Ryan Dahl was inspired to create Node.js after seeing a file upload progress bar on Flickr.

source: https://en.wikipedia.org/wiki/Node.js

By 2011

- Node.JS 0.1.14
- Package Management (npm)

Node.JS Rapid Adoption

LinkedIn

UBER

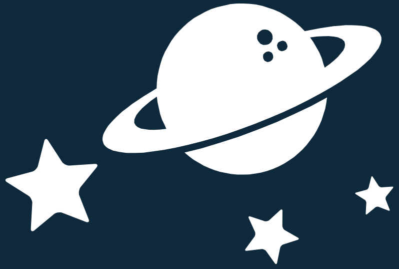
Walmart

ebay

PayPal

Node.JS is JavaScript

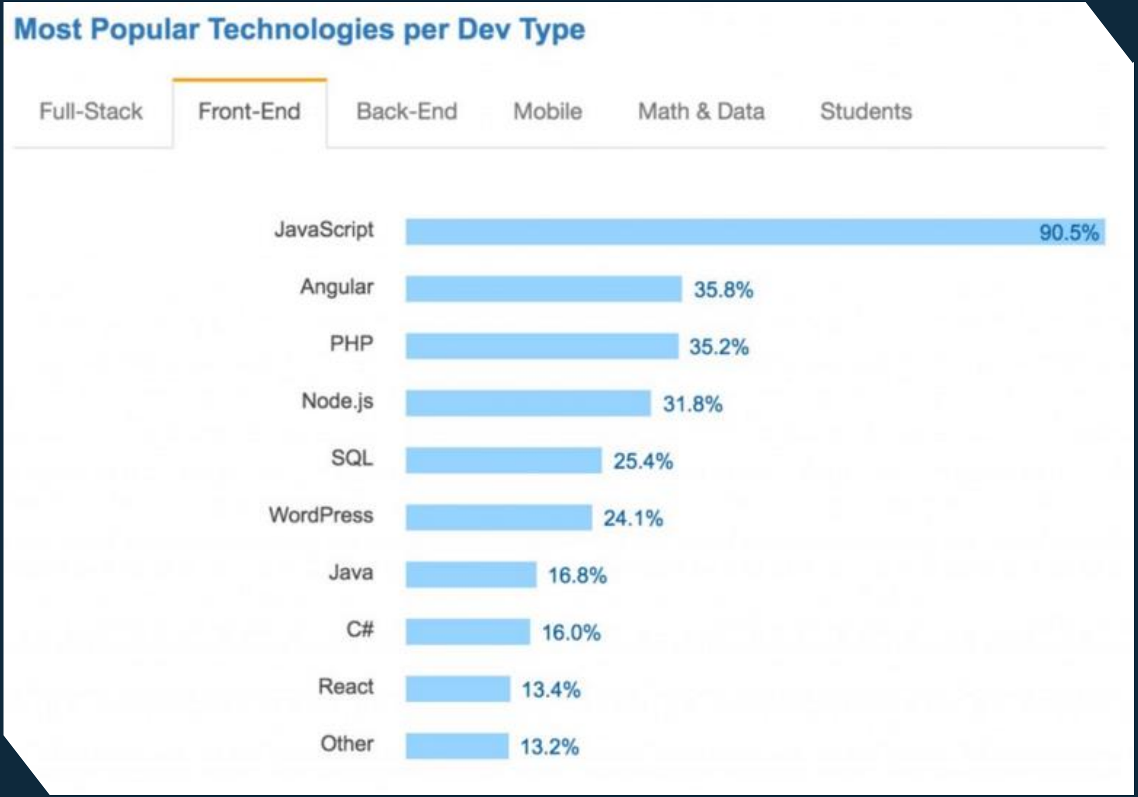
JavaScript is Everywhere



2015 GitHub Developer Survey

50,000 World Wide
Software Engineers

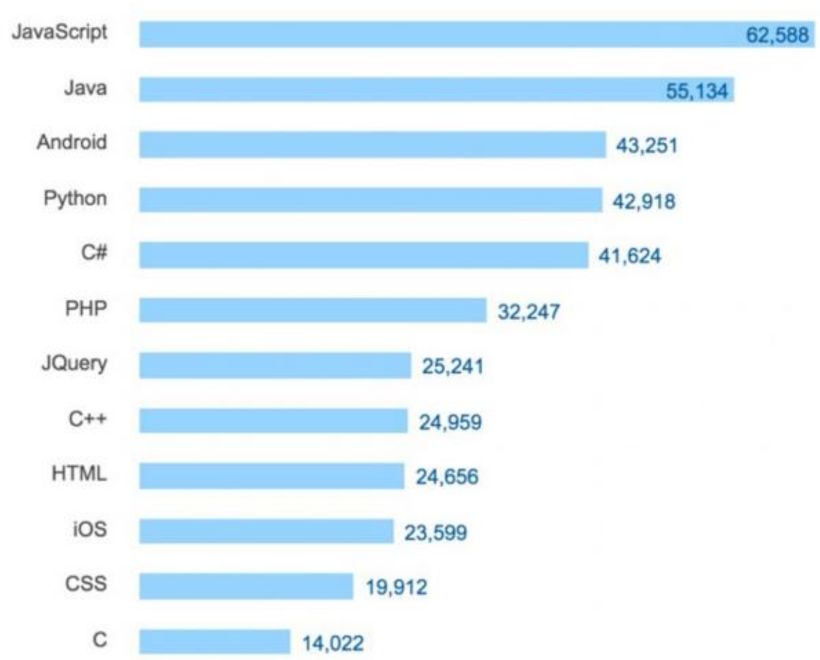
JavaScript wins Backend and Frontend popularity



Most Popular Technologies per Dev Type

JavaScript wins most open source projects

III. Top Tech on Stack Overflow



Security Horror Stories in Node.JS

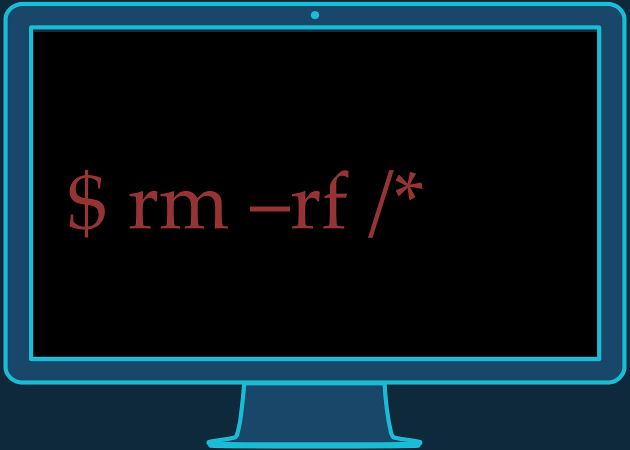
By January 2015

- rimrafall package published to npm

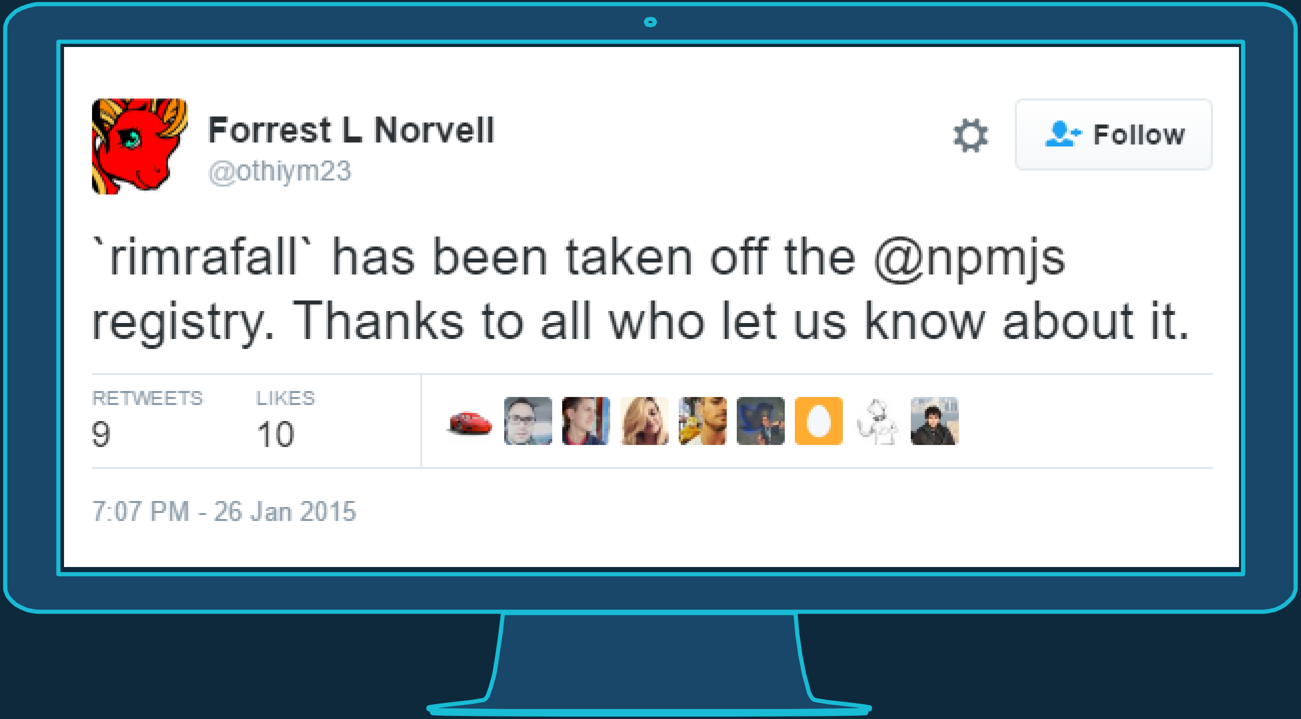
rimrafall ?

rimrafall

- npm pre-install script



```
$ rm –rf /*
```



Forrest L Norvell
@othiym23

`rimrafall` has been taken off the @npmjs registry. Thanks to all who let us know about it.

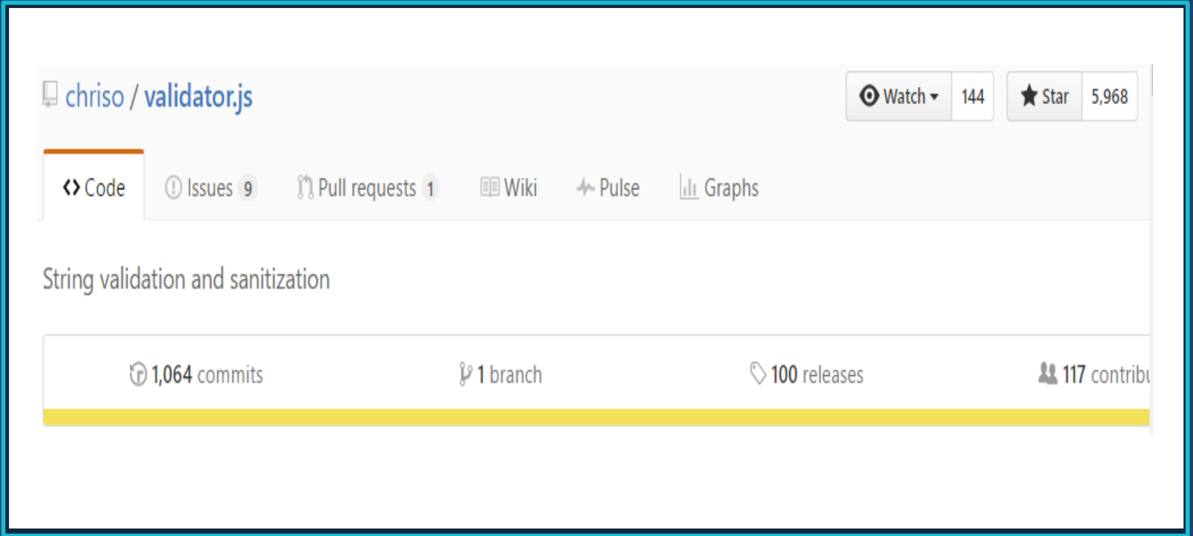
RETWEETS 9 LIKES 10

7:07 PM - 26 Jan 2015

Fishing Attacks, *npm* Style

validator.js

- helps validate and sanitize strings



chriso / validator.js

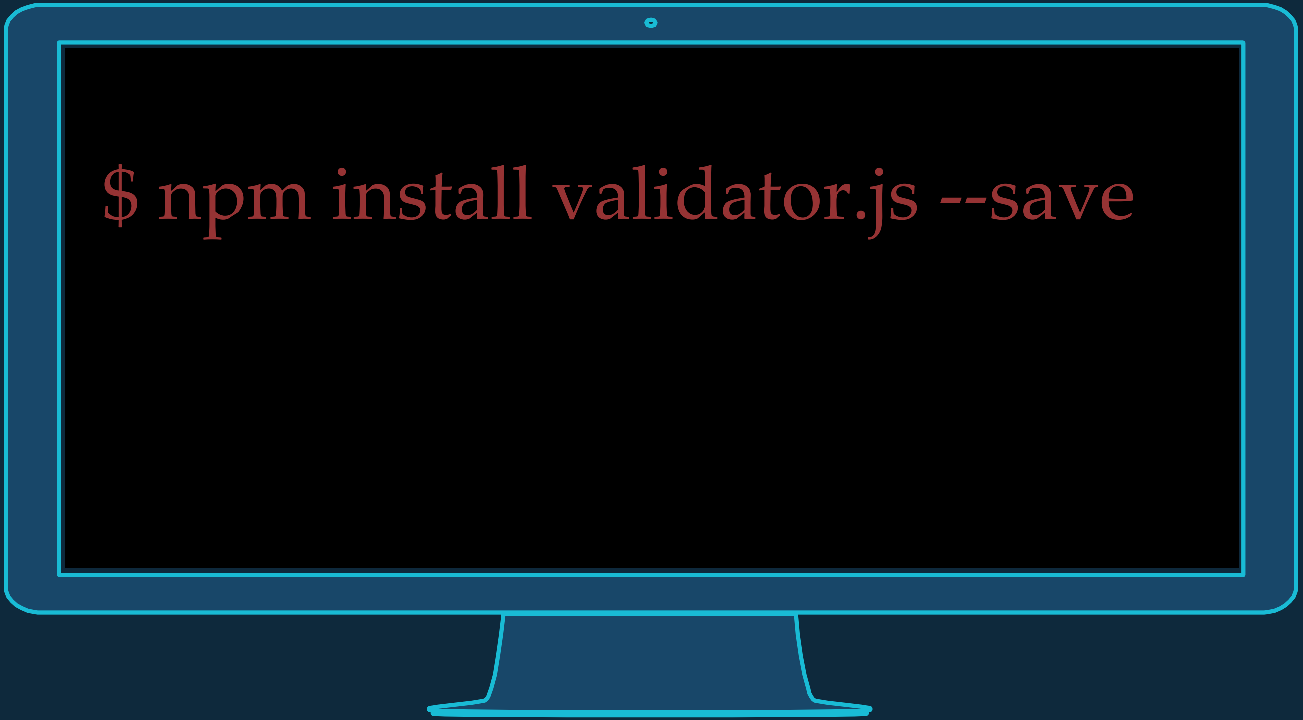
String validation and sanitization

1,064 commits

1 branch

100 releases

117 contribu



```
$ npm install validator.js --save
```

validator.js
!=
validator

malicious modules
of similar names

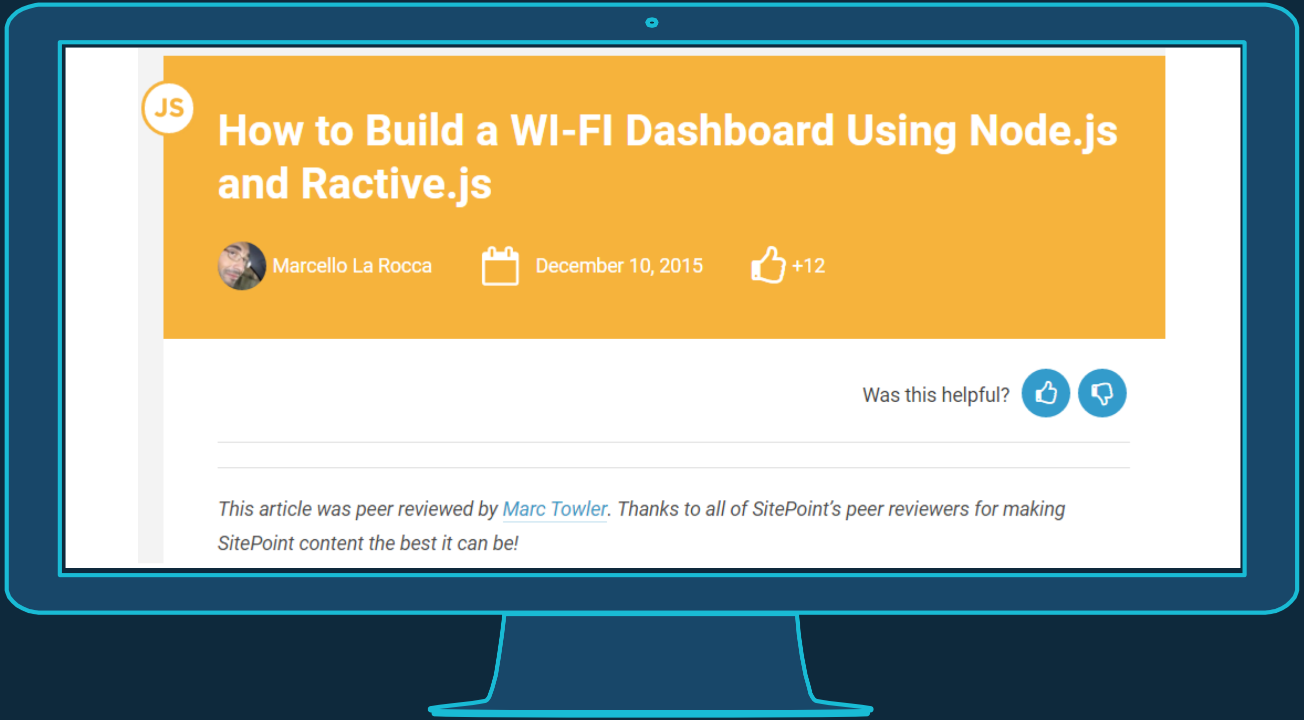
malicious modules
of similar names

3,500,000	socket.io
2,000	socketio

malicious modules
of similar names

11,000,000	uglify-js
50,000	uglifyjs

Failing to educate the younger generation



JS

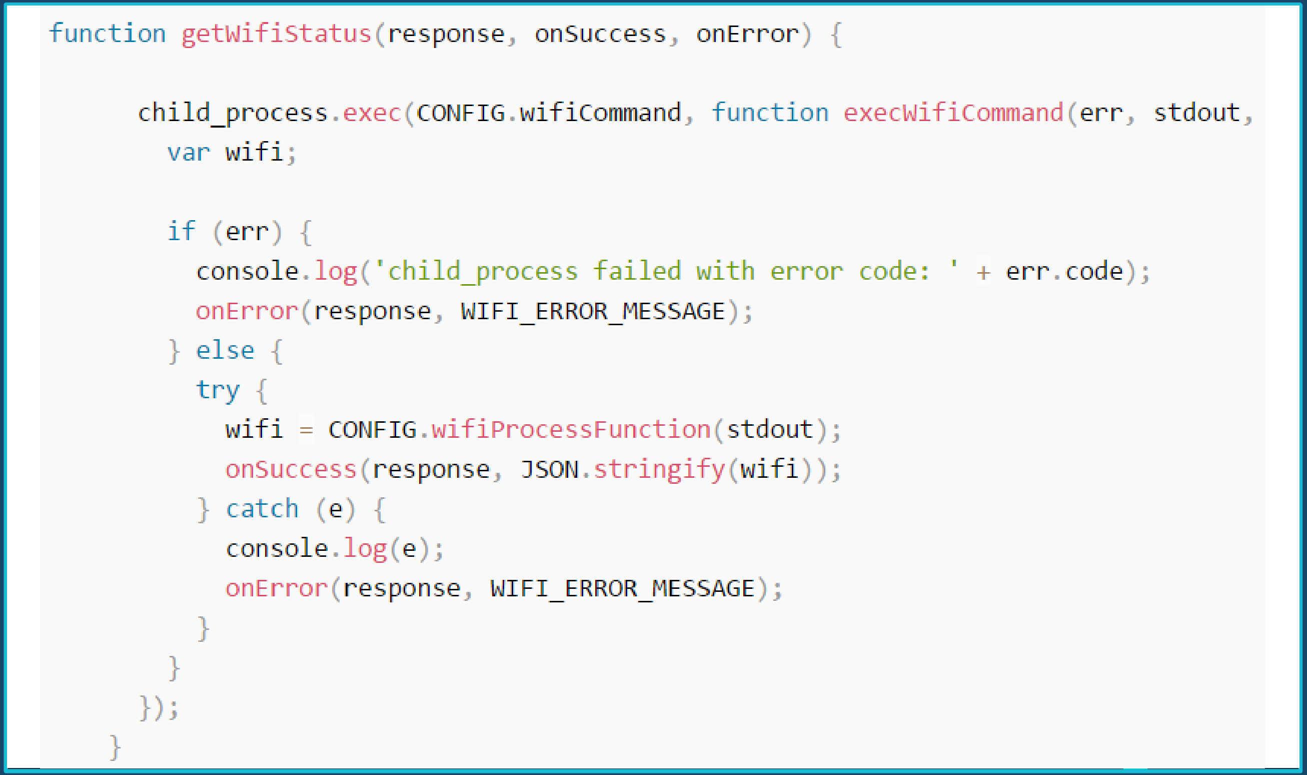
How to Build a WI-FI Dashboard Using Node.js and Ractive.js

Marcello La Rocca December 10, 2015 +12

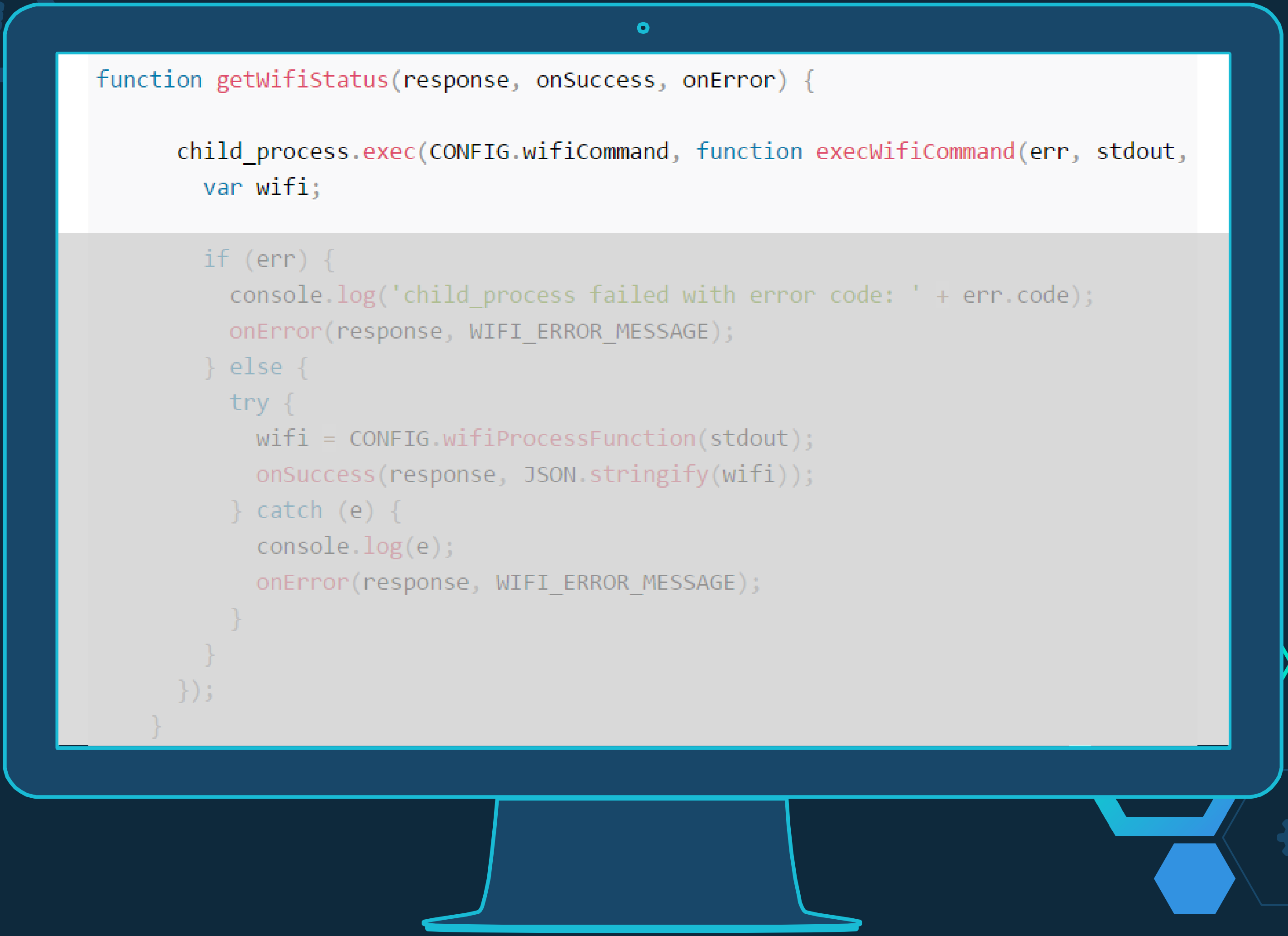
Was this helpful?

This article was peer reviewed by Marc Towler. Thanks to all of SitePoint's peer reviewers for making SitePoint content the best it can be!

seemingly innocent
tutorial to learn from



```
function getWifiStatus(response, onSuccess, onError) {

      child_process.exec(CONFIG.wifiCommand, function execWifiCommand(err, stdout,
        var wifi;

        if (err) {
          console.log('child_process failed with error code: ' + err.code);
          onError(response, WIFI_ERROR_MESSAGE);
        } else {
          try {
            wifi = CONFIG.wifiProcessFunction(stdout);
            onSuccess(response, JSON.stringify(wifi));
          } catch (e) {
            console.log(e);
            onError(response, WIFI_ERROR_MESSAGE);
          }
        }
      });
    }
```



```
function getWifiStatus(response, onSuccess, onError) {

      child_process.exec(CONFIG.wifiCommand, function execWifiCommand(err, stdout,
        var wifi;

        if (err) {
          console.log('child_process failed with error code: ' + err.code);
          onError(response, WIFI_ERROR_MESSAGE);
        } else {
          try {
            wifi = CONFIG.wifiProcessFunction(stdout);
            onSuccess(response, JSON.stringify(wifi));
          } catch (e) {
            console.log(e);
            onError(response, WIFI_ERROR_MESSAGE);
          }
        }
      });
    }
```

Tips & Recipes to Secure Node.js

1

Security by HTTP Headers

STRICT-TRANSPORT-SECURITY

Browsers enforce secure connections to the server (HTTPS)

X-FRAME-OPTIONS

Clickjacking protection by not rendering content in iframes

CONTENT-SECURITY-POLICY

Whitelist trusted content, and services

X-XSS-PROTECTION

enables *browser XSS filtering

*** IE8 | IE9**

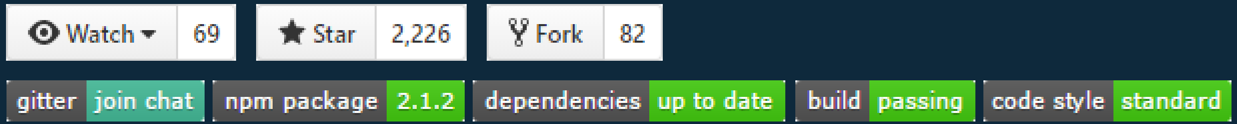
X-CONTENT-TYPE-OPTIONS

***browsers do not sniff MIME responses**

***IE8 | Chrome | Safari**

Helmet

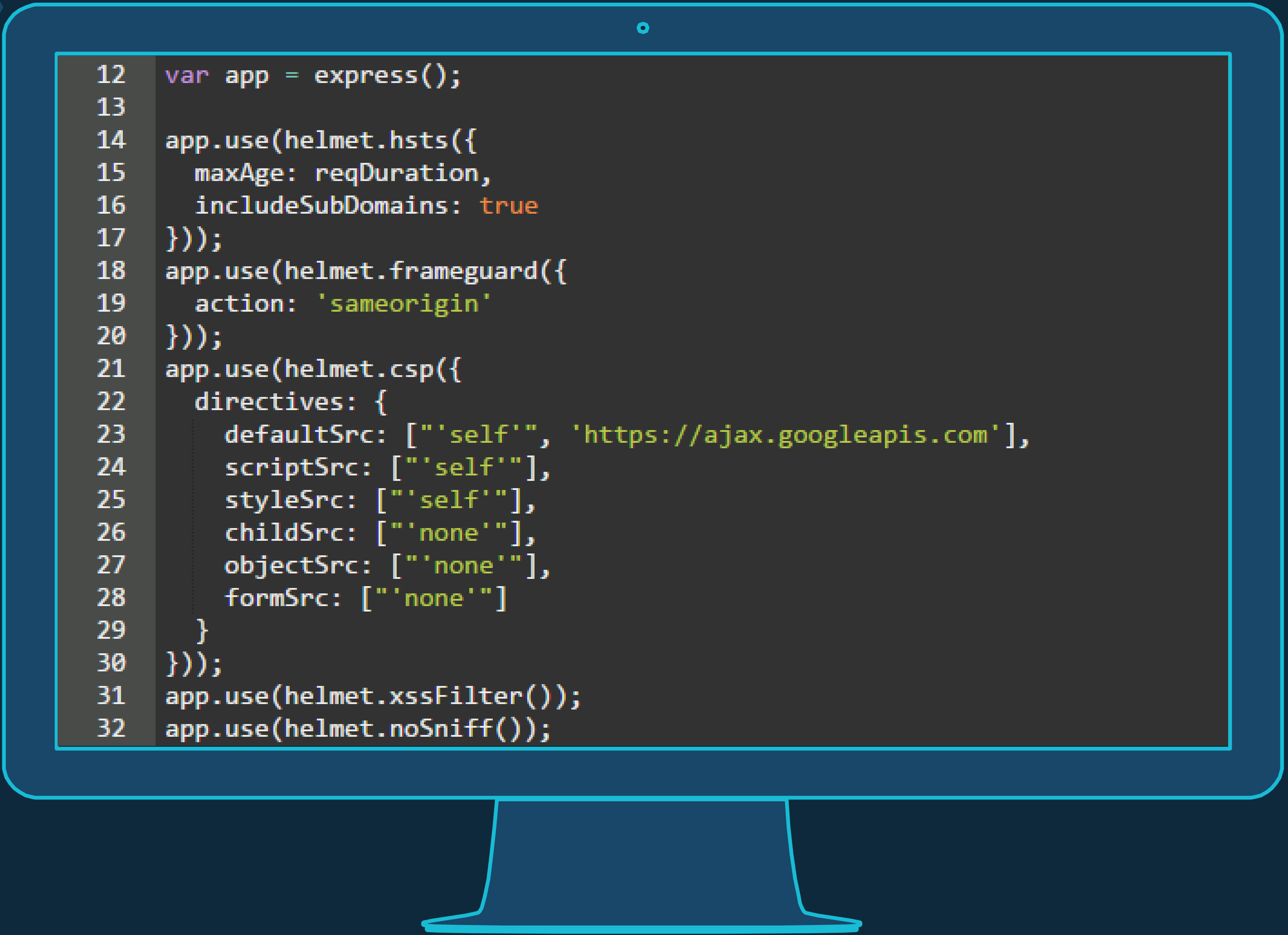
Securing ExpressJS



Watch 69 | Star 2,226 | Fork 82

gitter join chat | npm package 2.1.2 | dependencies up to date | build passing | code style standard

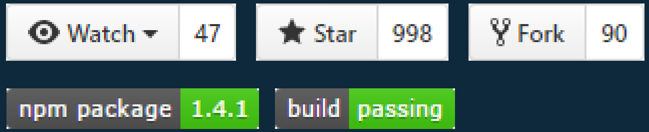
Putting it all together with Helmet and ExpressJS



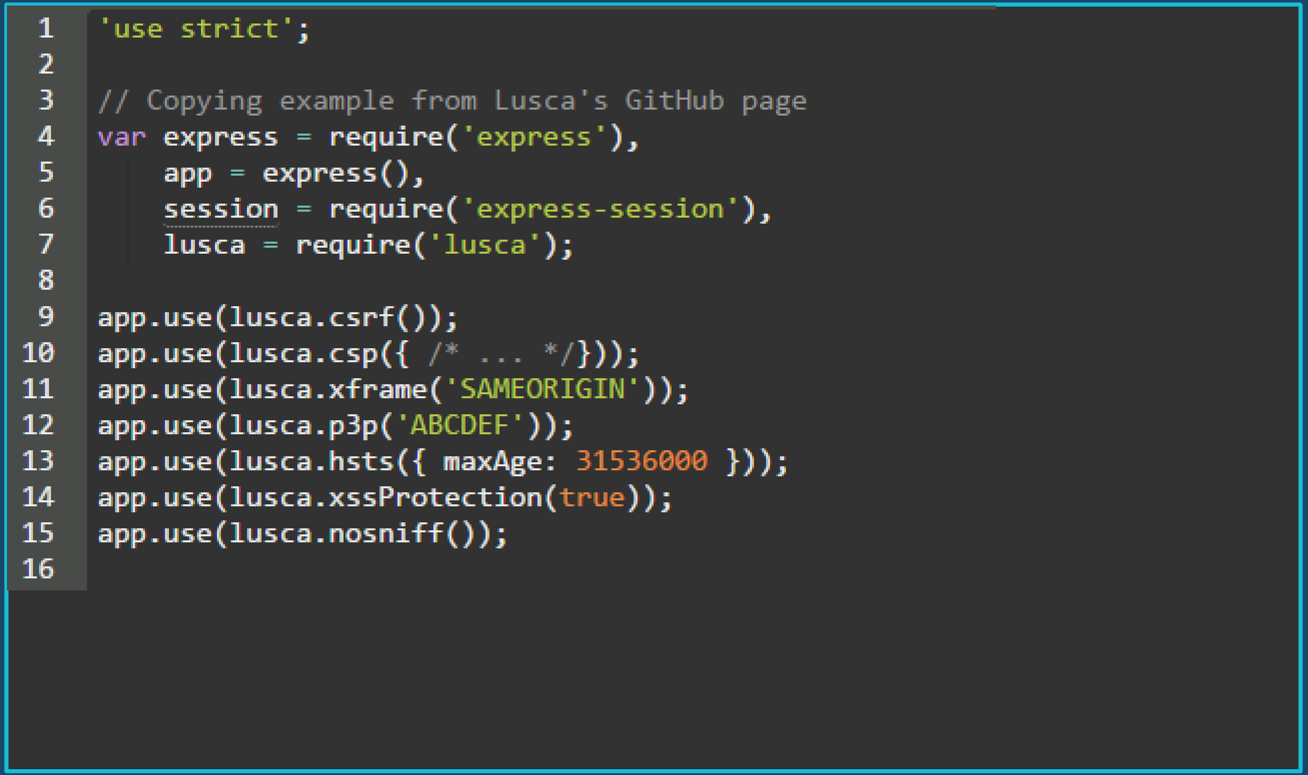
```
var app = express();

app.use(helmet.hsts({
  maxAge: reqDuration,
  includeSubDomains: true
}));
app.use(helmet.frameguard({
  action: 'sameorigin'
}));
app.use(helmet.csp({
  directives: {
    defaultSrc: ["'self'", 'https://ajax.googleapis.com'],
    scriptSrc: ["'self'"],
    styleSrc: ["'self'"],
    childSrc: ["'none'"],
    objectSrc: ["'none'"],
    formSrc: ["'none'"]
  }
}));
app.use(helmet.xssFilter());
app.use(helmet.noSniff());
```

Lusca

Securing ExpressJS



Putting it all together with Lusca and ExpressJS



```
'use strict';

// Copying example from Lusca's GitHub page
var express = require('express'),
    app = express(),
    session = require('express-session'),
    lusca = require('lusca');

app.use(lusca.csrf());
app.use(lusca.csp({ /* ... */}));
app.use(lusca.xframe('SAMEORIGIN'));
app.use(lusca.p3p('ABCDEF'));
app.use(lusca.hsts({ maxAge: 31536000 }));
app.use(lusca.xssProtection(true));
app.use(lusca.nosniff());
```

2

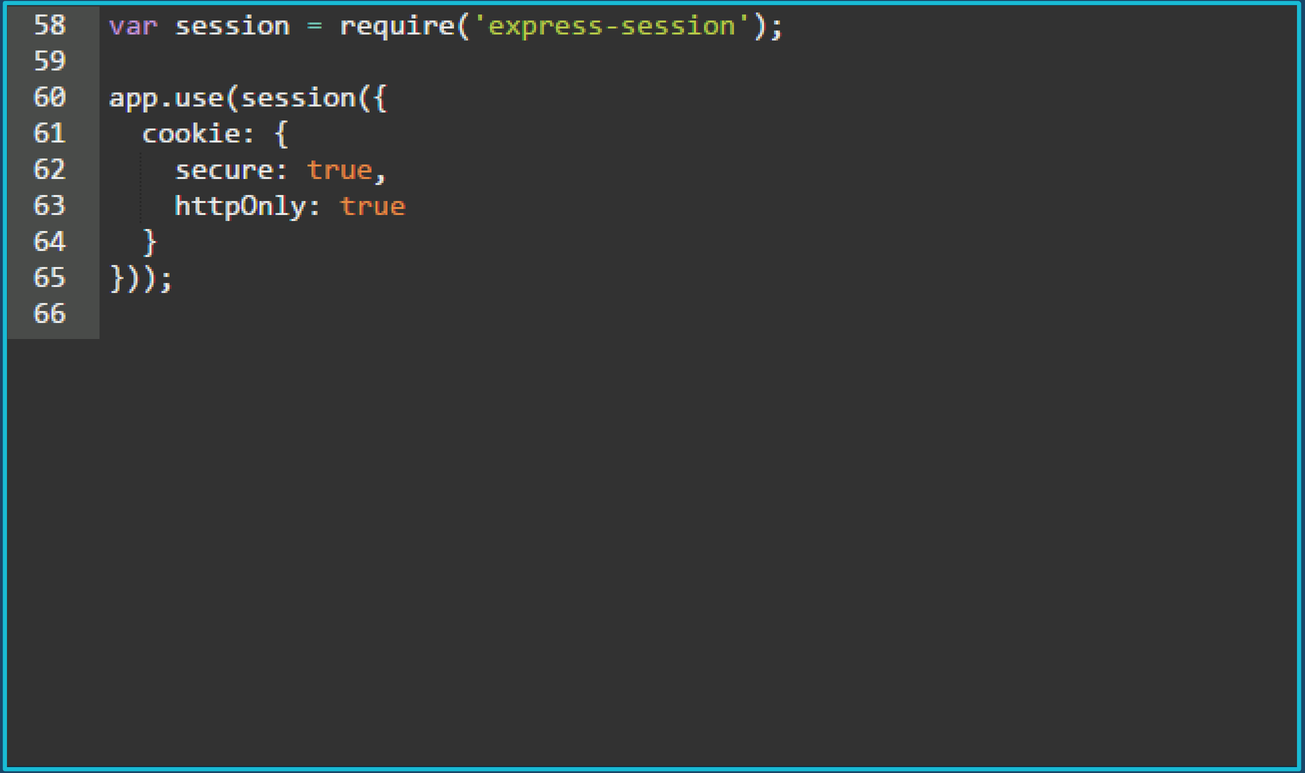
Securing the Cookies

SECURE

cookies sent over HTTPS connections only

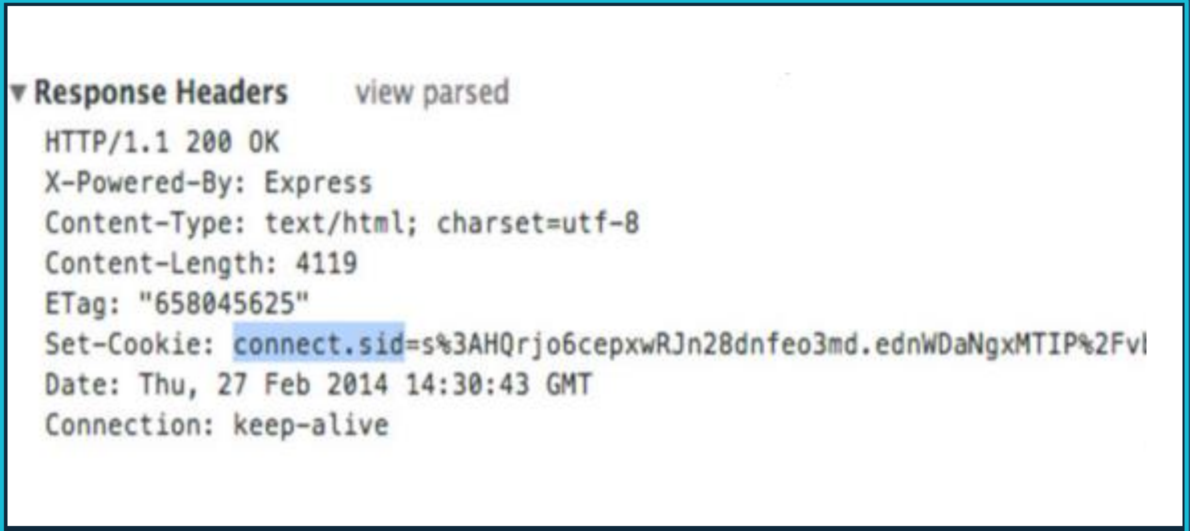
httpOnly

cookies are not accessible from JavaScript

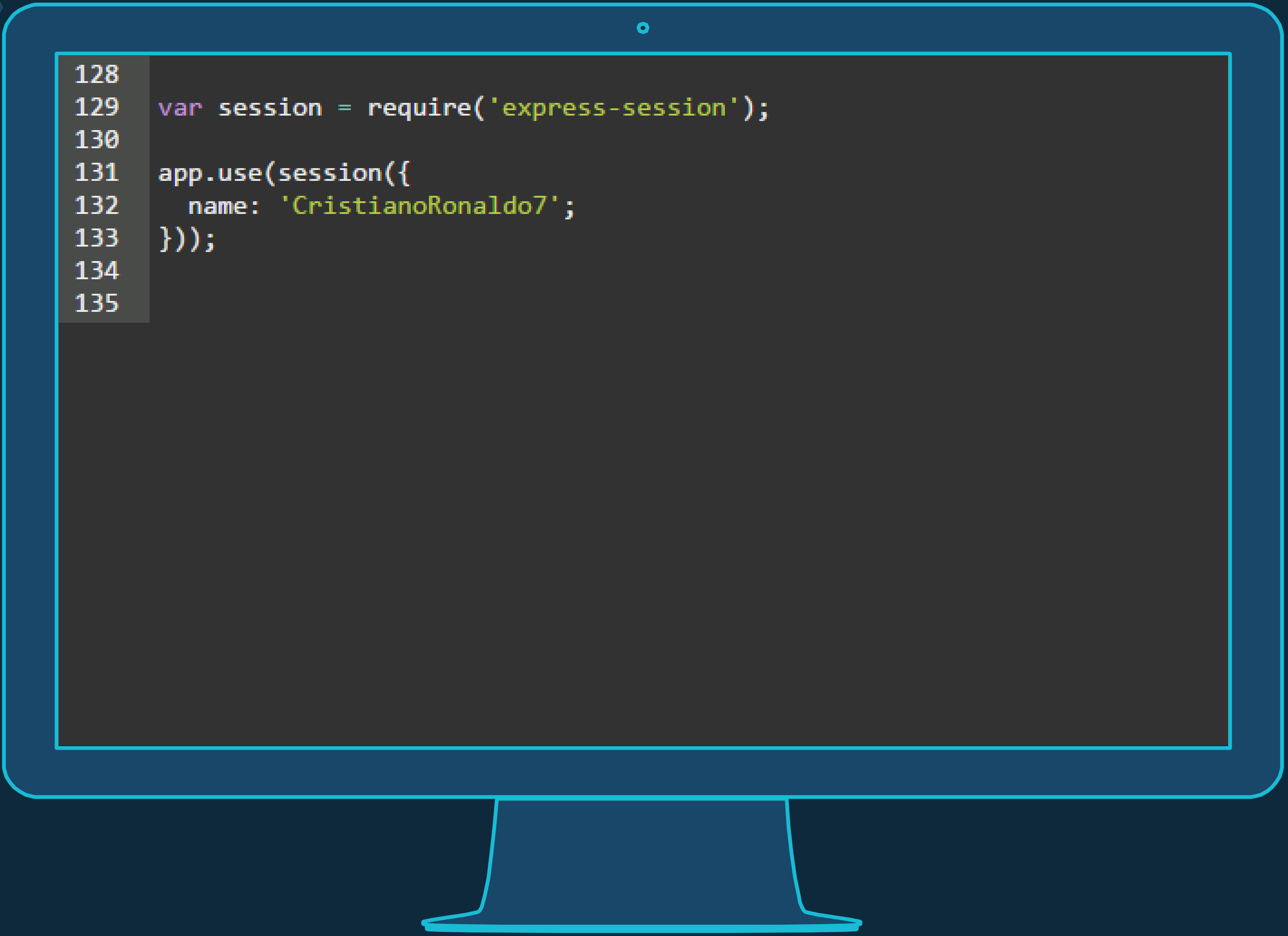


```
var session = require('express-session');

app.use(session({
  cookie: {
    secure: true,
    httpOnly: true
  }
}));
```

Fingerprinting Node.JS

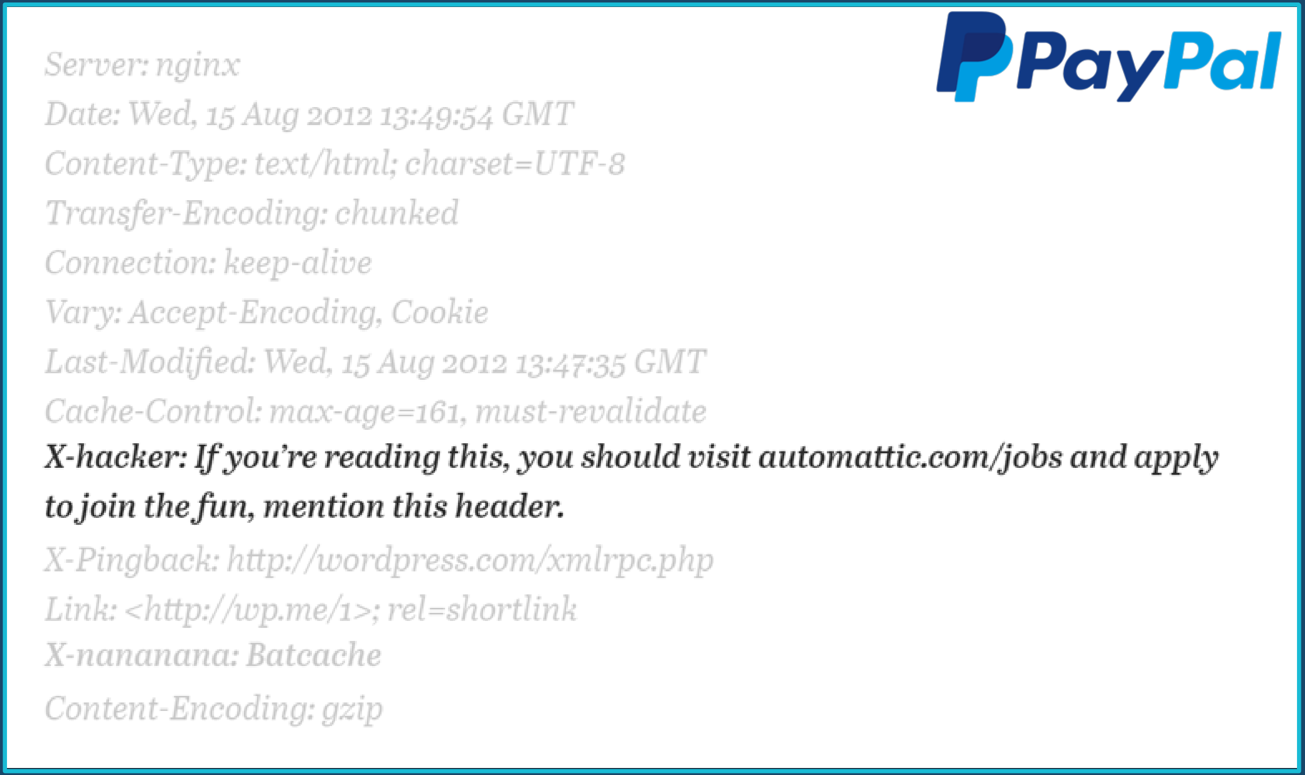


```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 4119
ETag: "658045625"
Set-Cookie: connect.sid=s%3AHQrjo6cepxwRJn28dnfeo3md.ednWDaNgxMTIP%2Fvl
Date: Thu, 27 Feb 2014 14:30:43 GMT
Connection: keep-alive
```

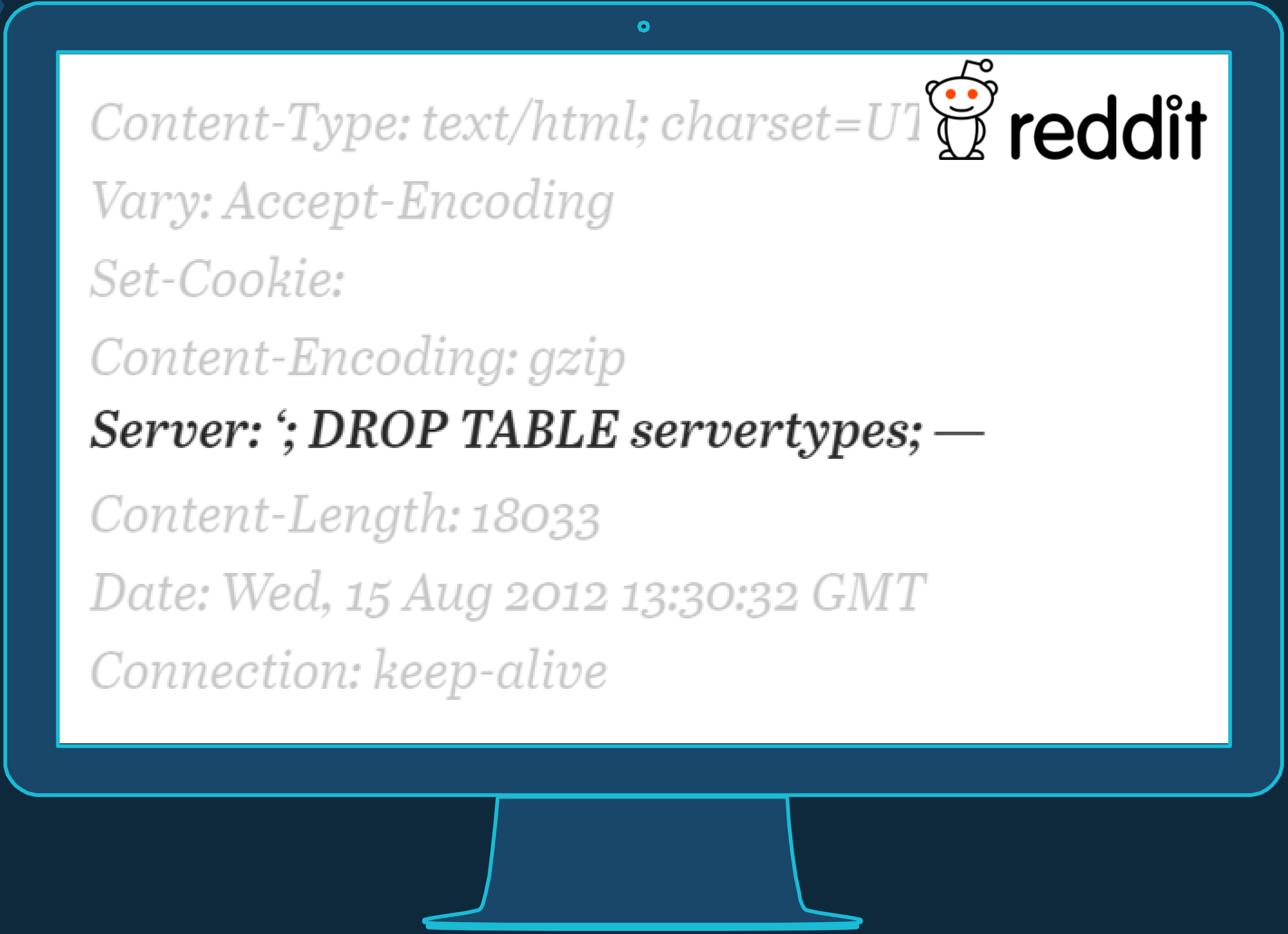


```
var session = require('express-session');

app.use(session({
  name: 'CristianoRonaldo7';
}));
```

Fun with Headers



```
Server: nginx
Date: Wed, 15 Aug 2012 13:49:54 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding, Cookie
Last-Modified: Wed, 15 Aug 2012 13:47:35 GMT
Cache-Control: max-age=161, must-revalidate
X-hacker: If you're reading this, you should visit automattic.com/jobs and apply
to join the fun, mention this header.
X-Pingback: http://wordpress.com/xmlrpc.php
Link: <http://wp.me/1>; rel=shortlink
X-nananana: Batcache
Content-Encoding: gzip
```

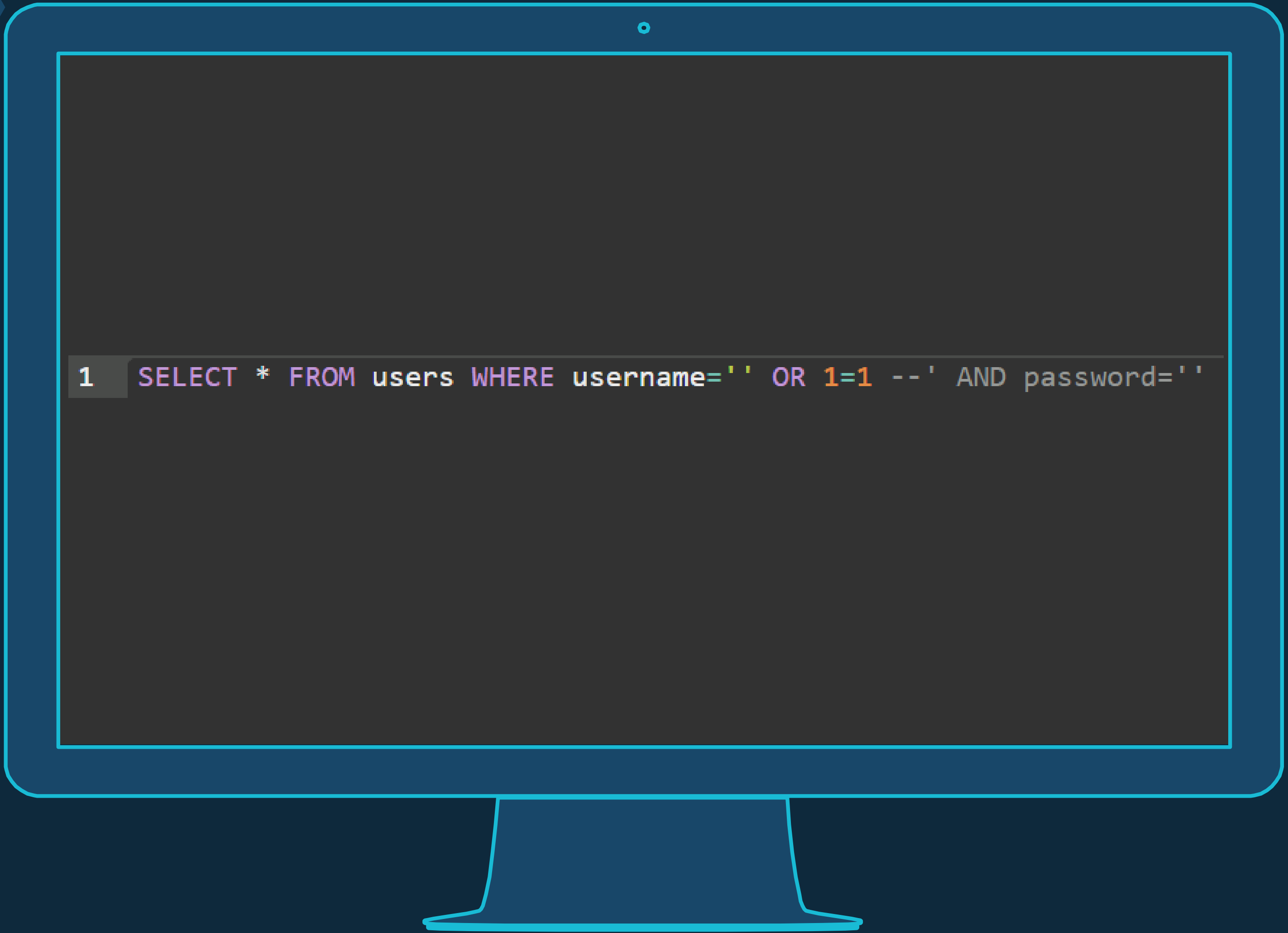


```
Content-Type: text/html; charset=UT
Vary: Accept-Encoding
Set-Cookie:
Content-Encoding: gzip
Server: ‘; DROP TABLE servertypes; —
Content-Length: 18033
Date: Wed, 15 Aug 2012 13:30:32 GMT
Connection: keep-alive
```

3

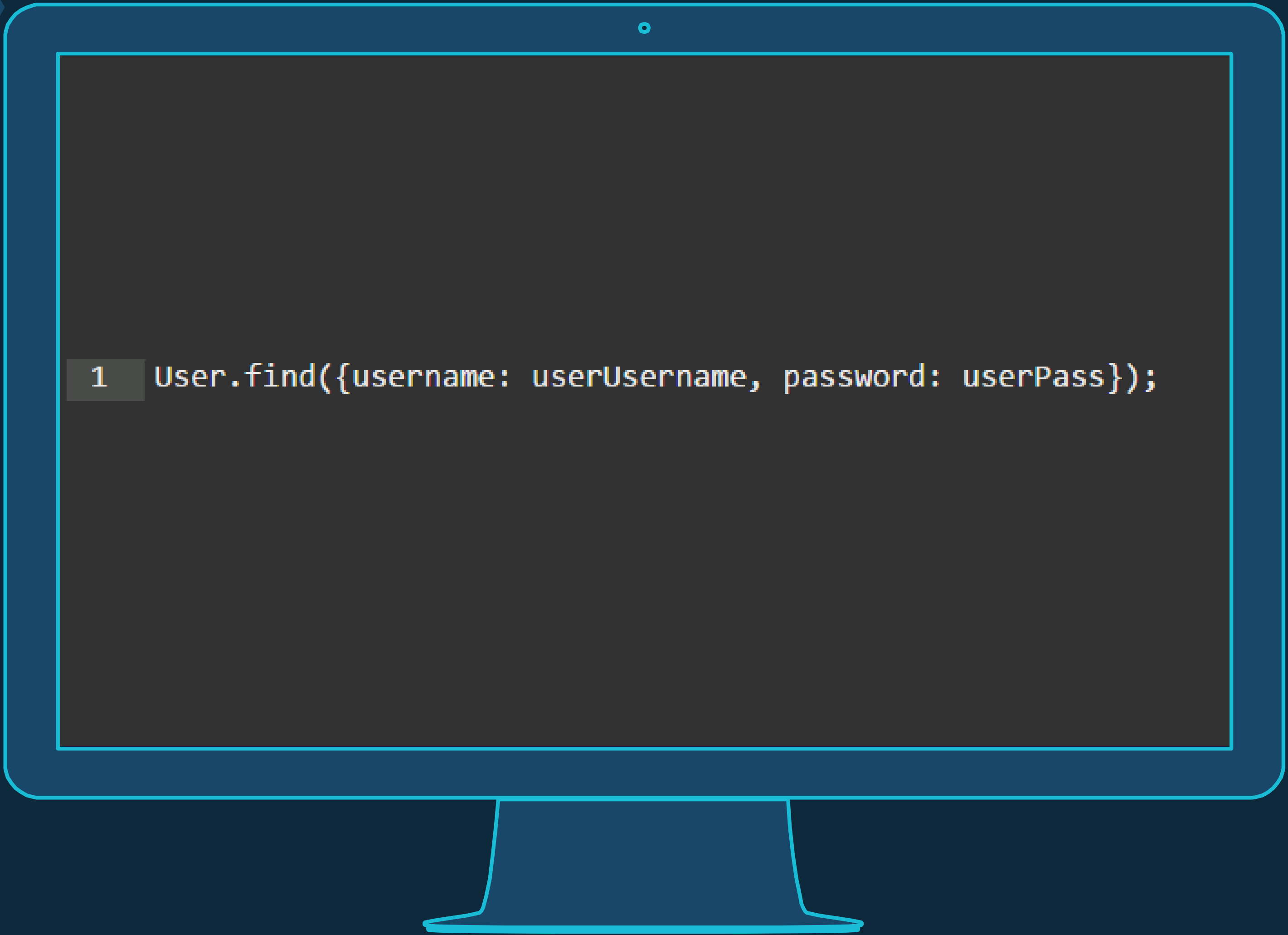
noSQL Injections

Creating TRUE SQL statements



```sql
SELECT * FROM users WHERE username='' OR 1=1 --' AND password=''
```

Creating TRUE SQL statements



```
User.find({username: userUsername, password: userPass});
```

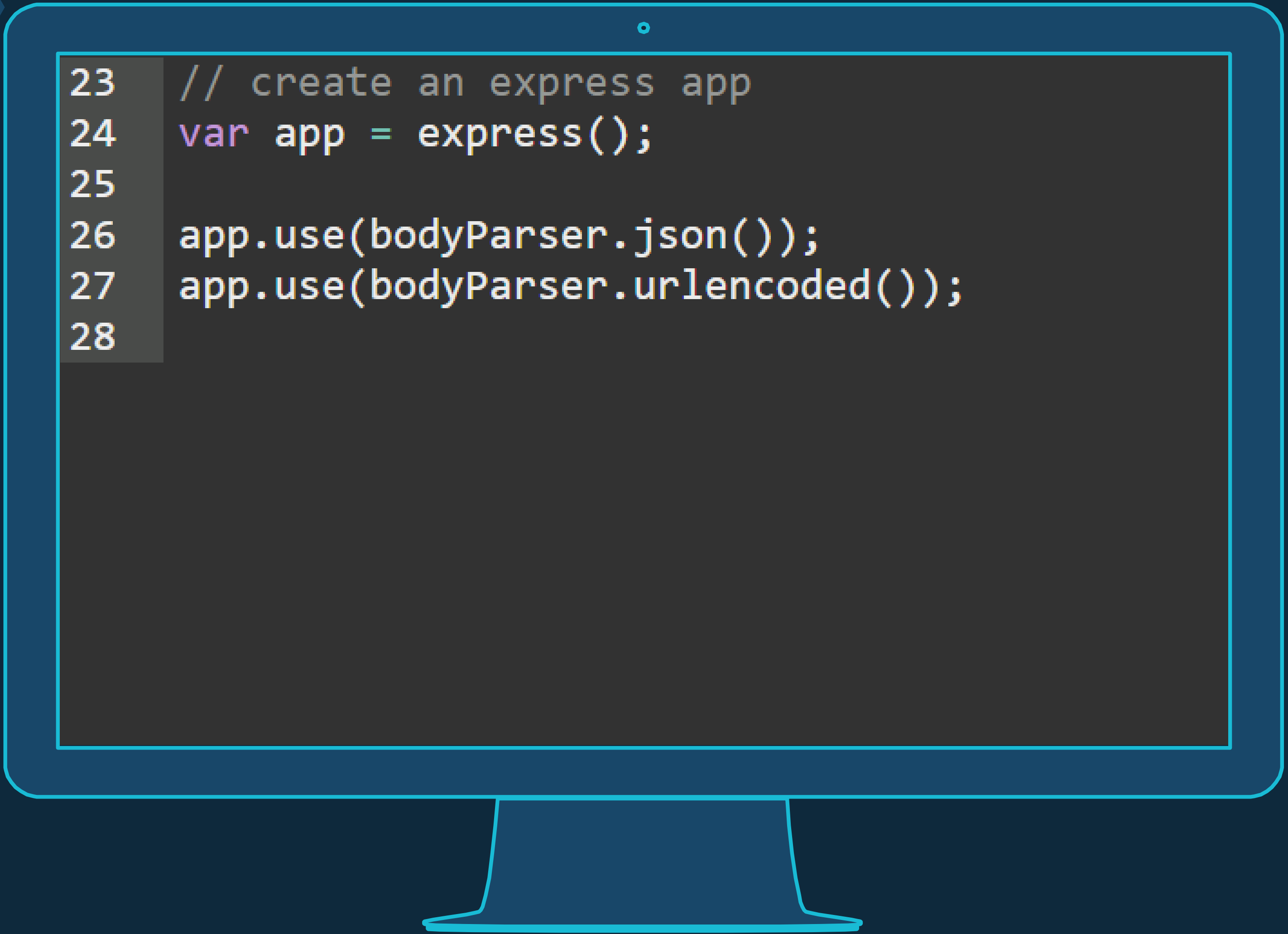
show me the code…

Live Demo!

No HTTP body in ExpressJS

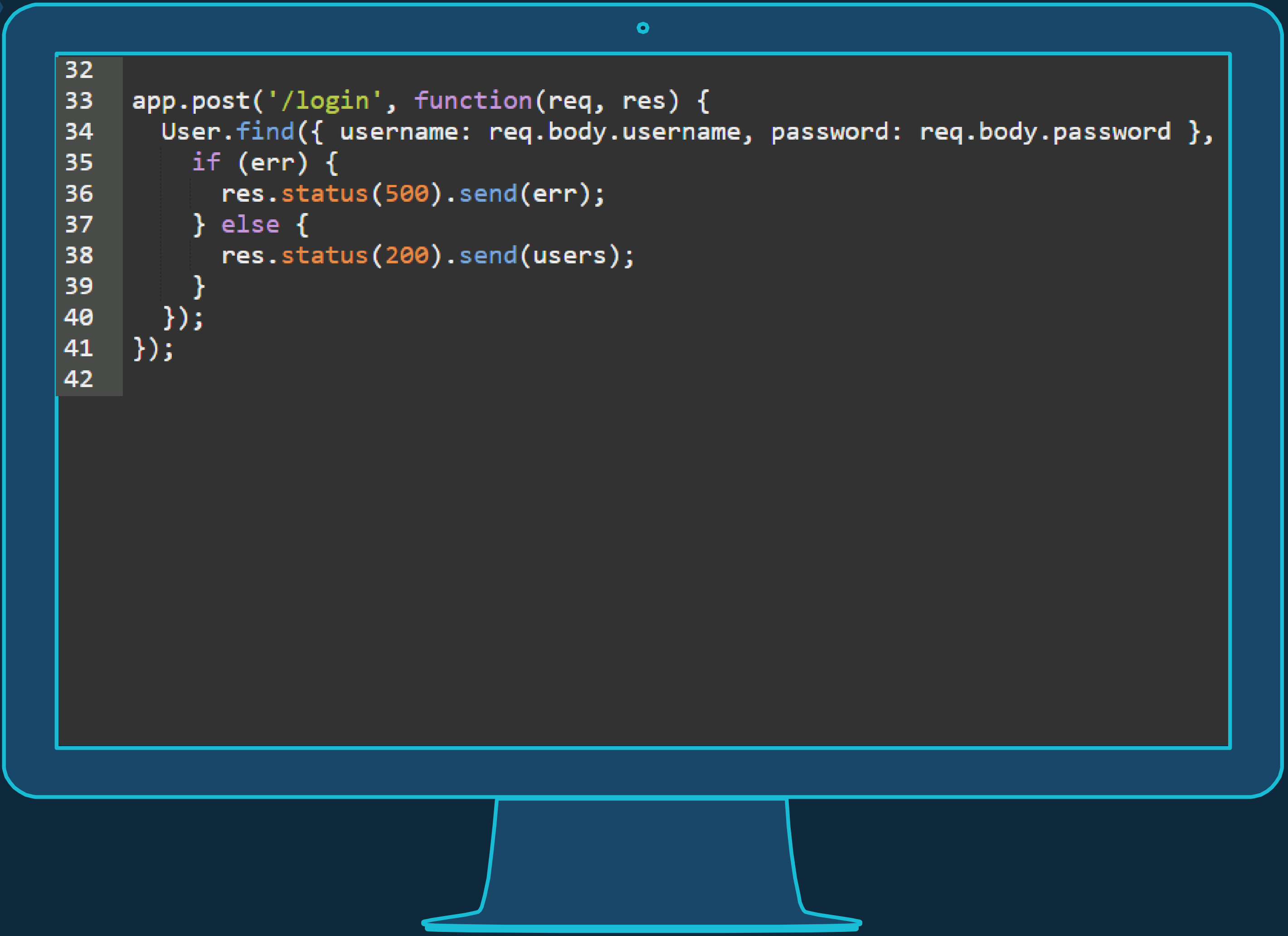
it relies on bodyParser lib

ExpressJS uses bodyParser library to access HTTP body payload



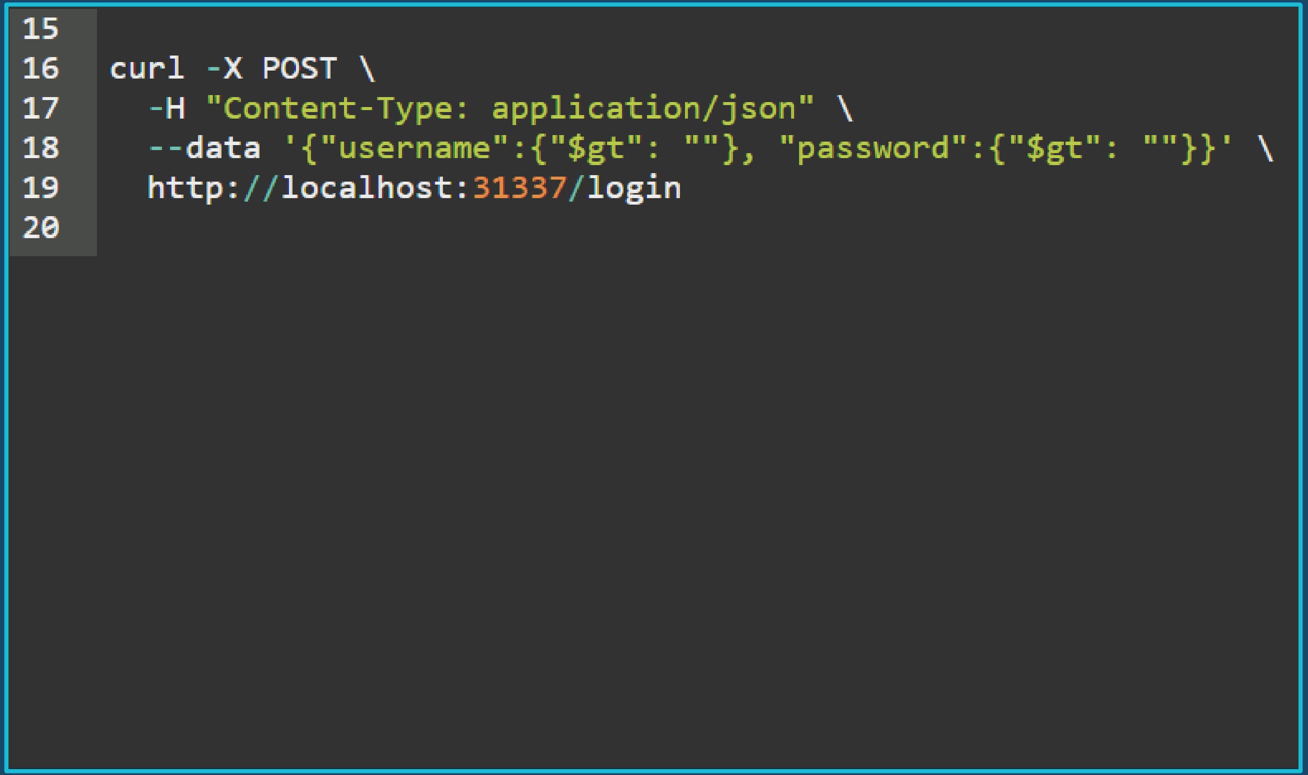
```
// create an express app
var app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded());

```

ExpressJS uses bodyParser library to access HTTP body payload



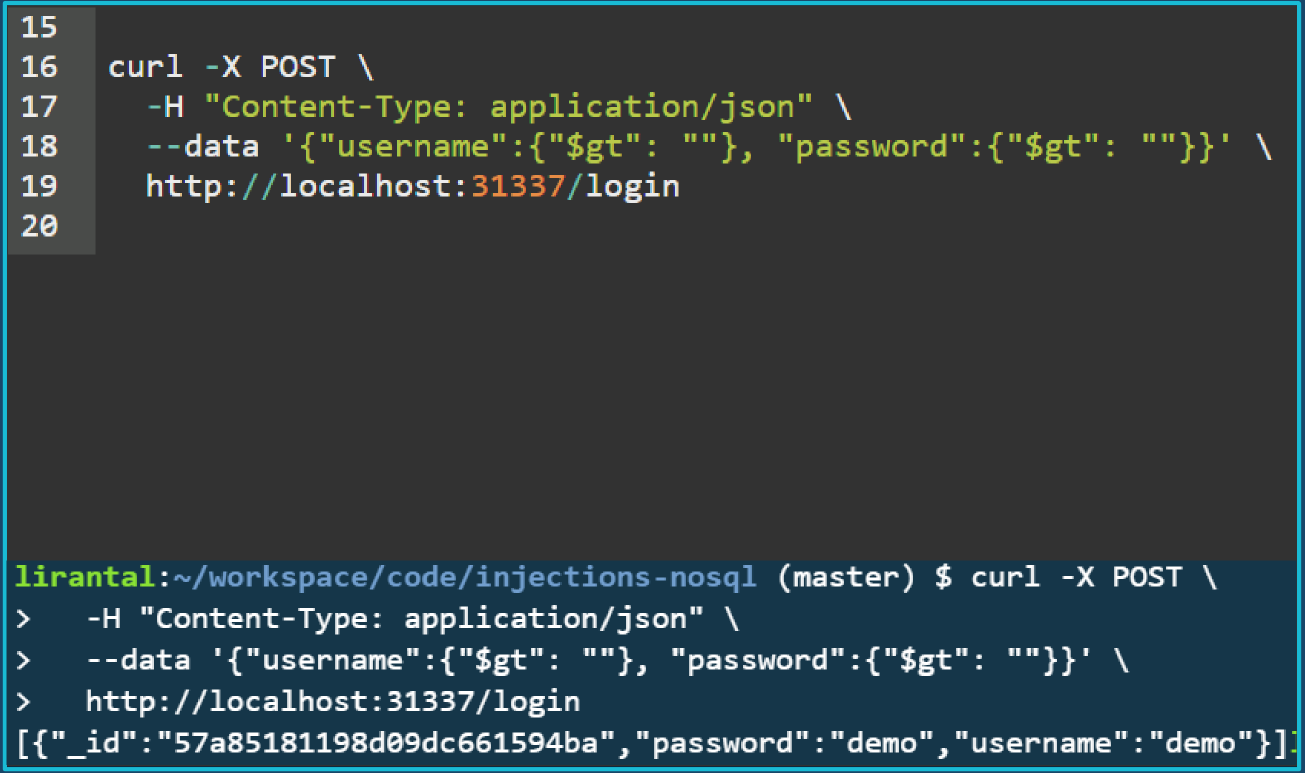
```
app.post('/login', function(req, res) {
  User.find({ username: req.body.username, password: req.body.password },
    if (err) {
      res.status(500).send(err);
    } else {
      res.status(200).send(users);
    }
  });
});
```

Creating TRUE SQL statements



```
curl -X POST \
  -H "Content-Type: application/json" \
  --data '{"username":{"$gt": ""}, "password":{"$gt": ""}}' \
  http://localhost:31337/login
```

Creating TRUE SQL statements



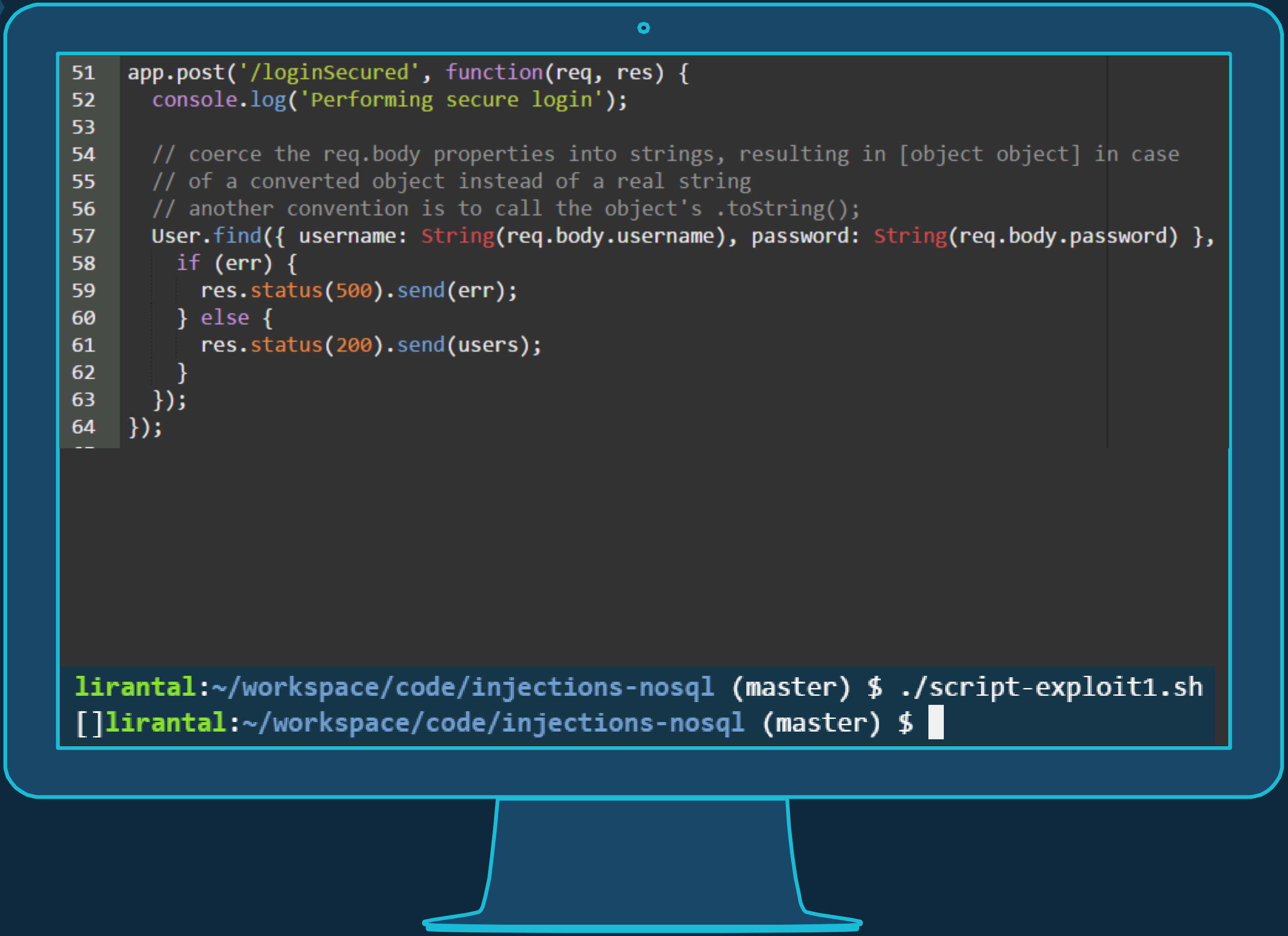
```
curl -X POST \
  -H "Content-Type: application/json" \
  --data '{"username":{"$gt": ""}, "password":{"$gt": ""}}' \
  http://localhost:31337/login
```

```
lirantal:~/workspace/code/injections-nosql (master) $ curl -X POST \
>   -H "Content-Type: application/json" \
>   --data '{"username":{"$gt": ""}, "password":{"$gt": ""}}' \
>   http://localhost:31337/login
[{"_id":"57a85181198d09dc661594ba","password":"demo","username":"demo"}]
```

Validate Input

- Validate Length and Type
- Validate & Sanitize input to expected type
- Parameters Binding
- Security in Depth

ExpressJS uses bodyParser library to access HTTP body payload



```
app.post('/loginSecured', function(req, res) {
  console.log('Performing secure login');

  // coerce the req.body properties into strings, resulting in [object object] in case
  // of a converted object instead of a real string
  // another convention is to call the object's .toString();
  User.find({ username: String(req.body.username), password: String(req.body.password) },
    if (err) {
      res.status(500).send(err);
    } else {
      res.status(200).send(users);
    }
  });
});
```

```
lirantal:~/workspace/code/injections-nosql (master) $ ./script-exploit1.sh
[]lirantal:~/workspace/code/injections-nosql (master) $ 
```

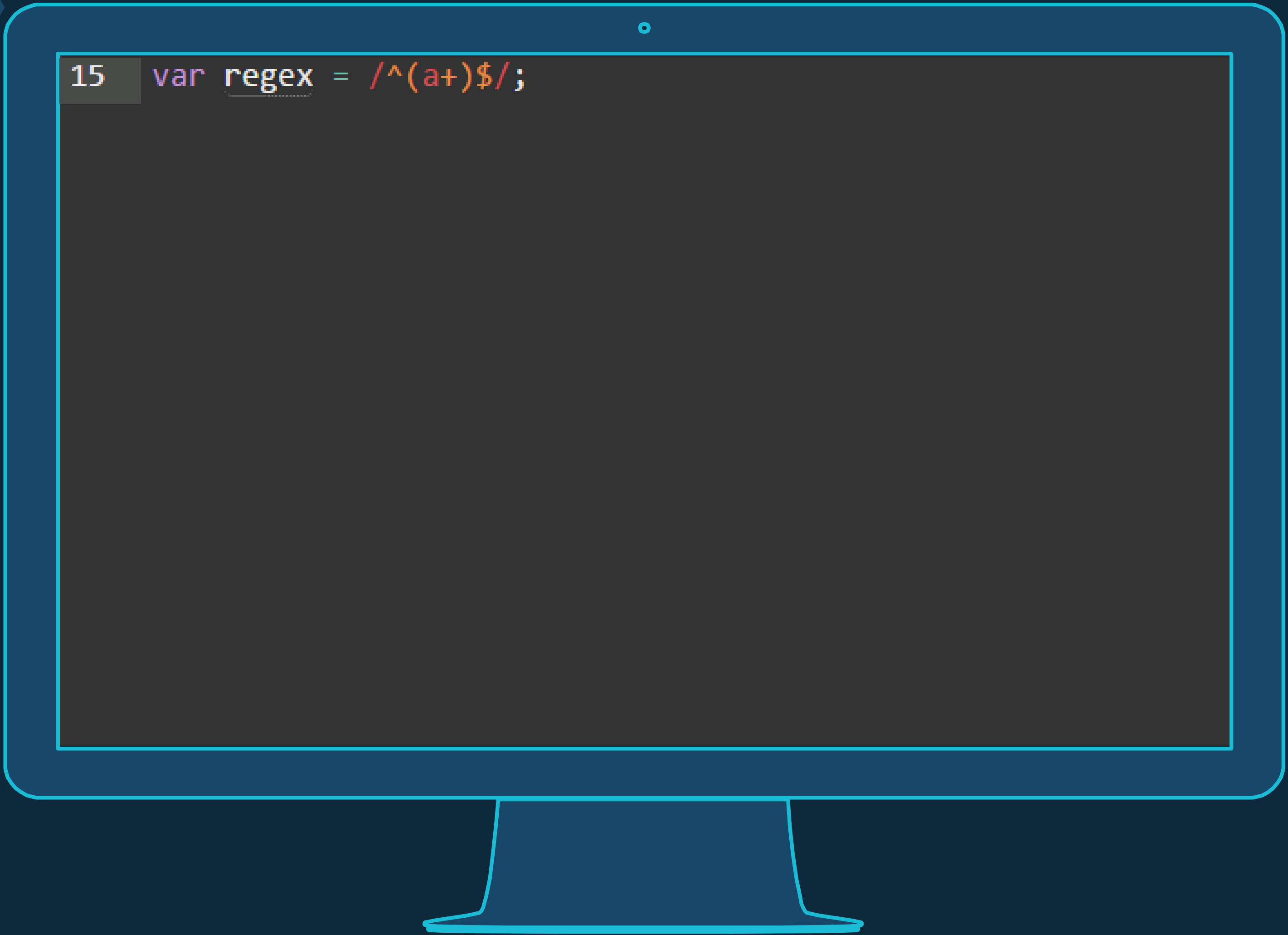
4

ReDoS

Regular Expressions DoS

Requirement:

- Validate the input has at least one ‘a’ character and allow unlimited occurences



```
var regex = /^(a+)$/;
```

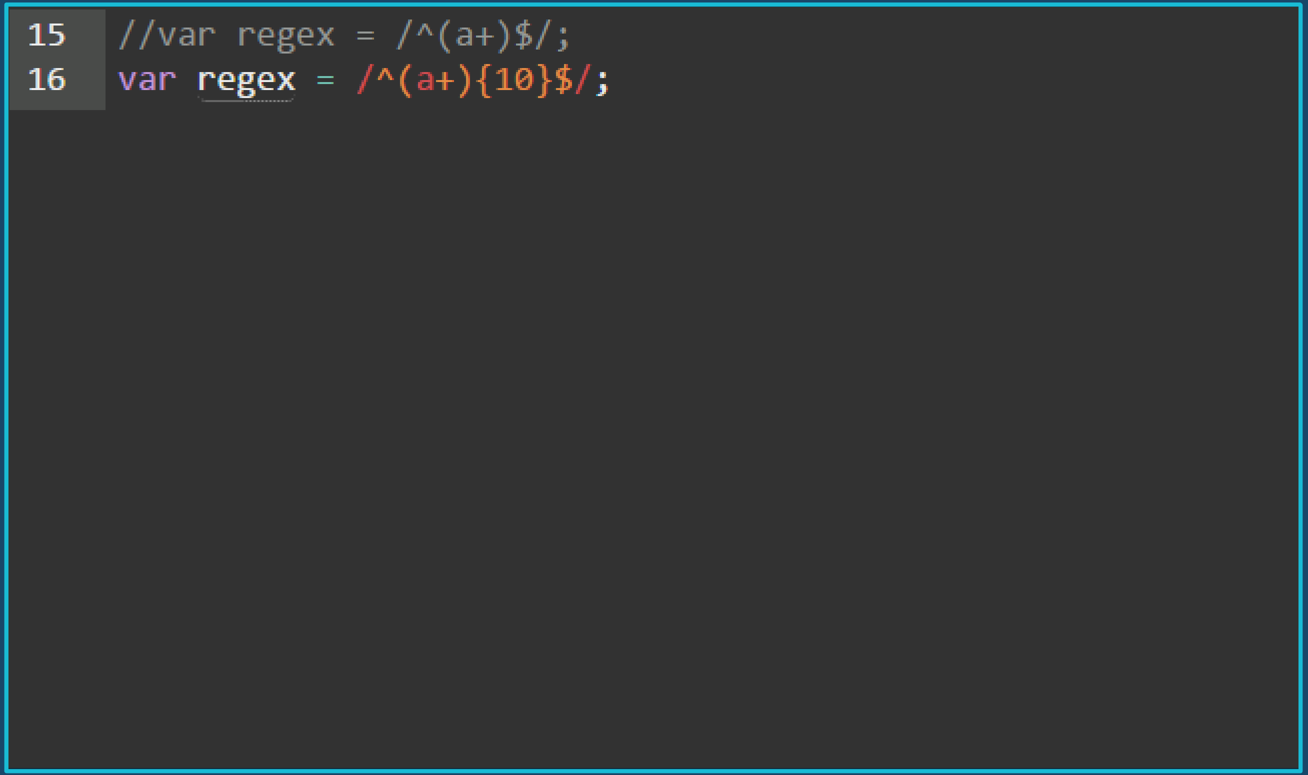
3 Months Later…

More work on the feature:

- Different Engineer gets the job
- Requirement changes: Validate the input has exactly 10 characters of ‘a’

show me the code…

Live Demo!



```
//var regex = /^(a+)$/;
var regex = /^(a+){10}$/;
```

Attacker sends

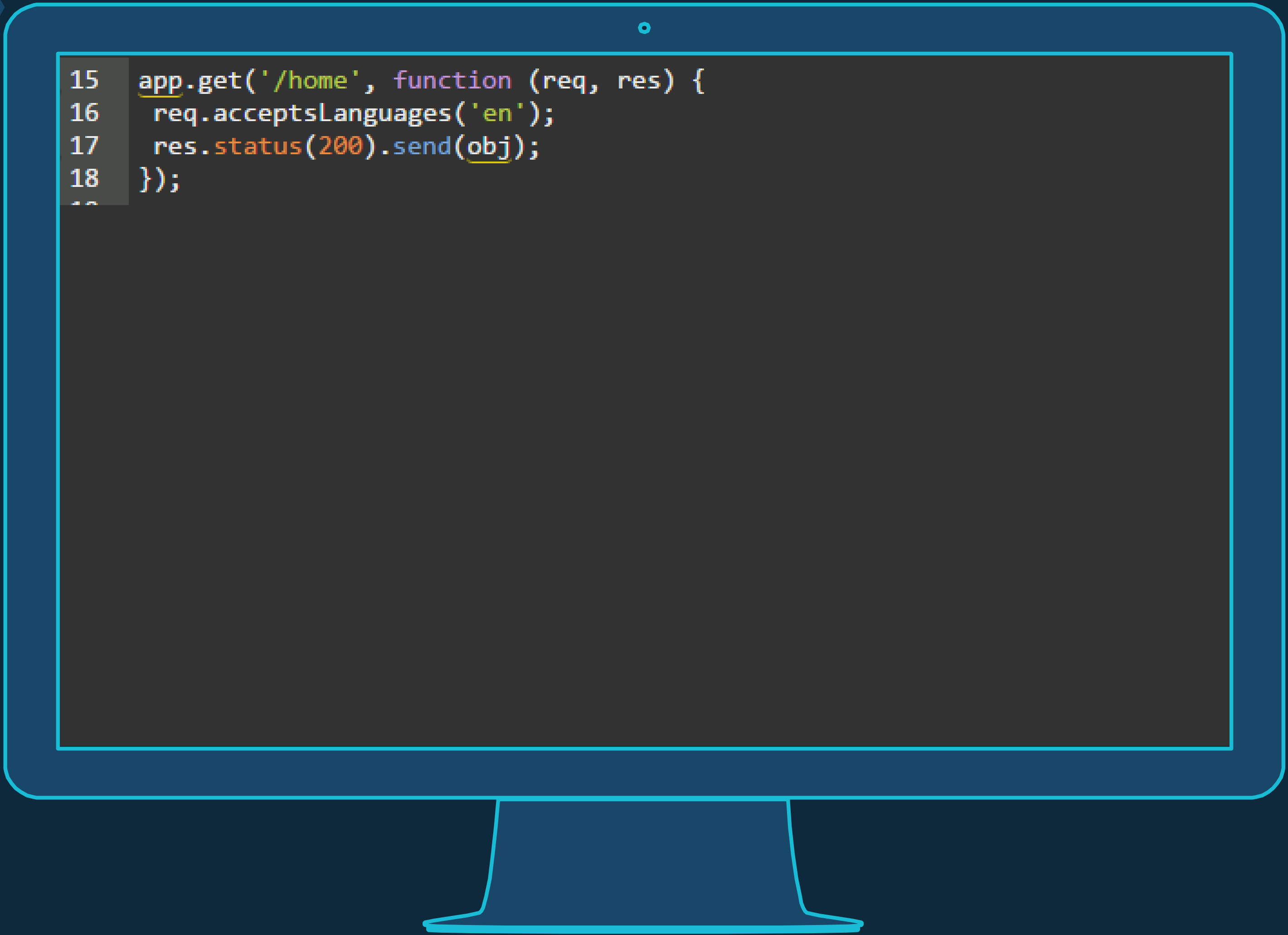
- Array(100).join(‘a’) + ‘!’

BOOM

ExpressJS uses neogitator

- ◇ **parsing the Accept-Language header**

Parameters Binding

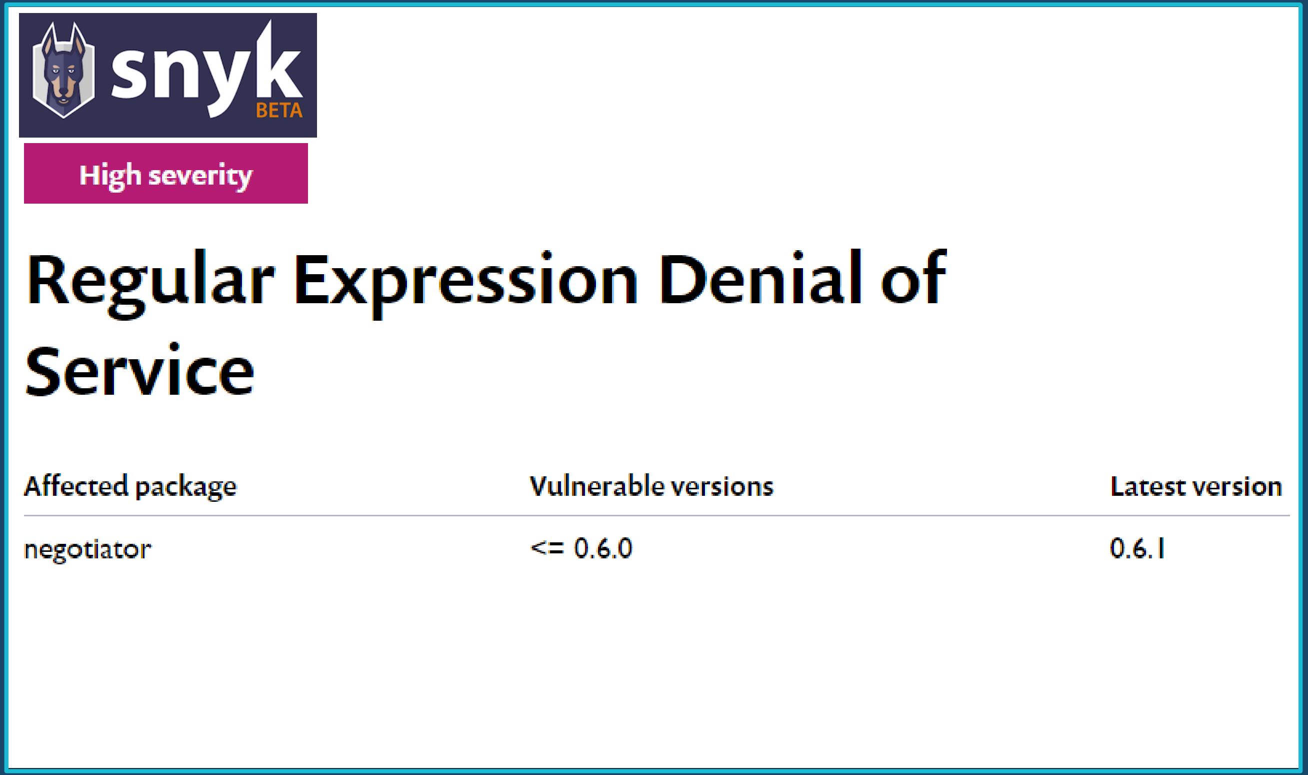


```
app.get('/home', function (req, res) {
 req.acceptsLanguages('en');
 res.status(200).send(obj);
});
```

10,000,000

negotiator

npm v0.6.1 | downloads 10M/month | node >= 0.6 | build passing | coverage 100%



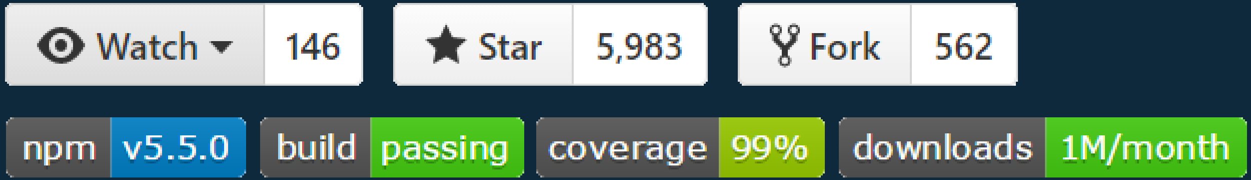
High severity

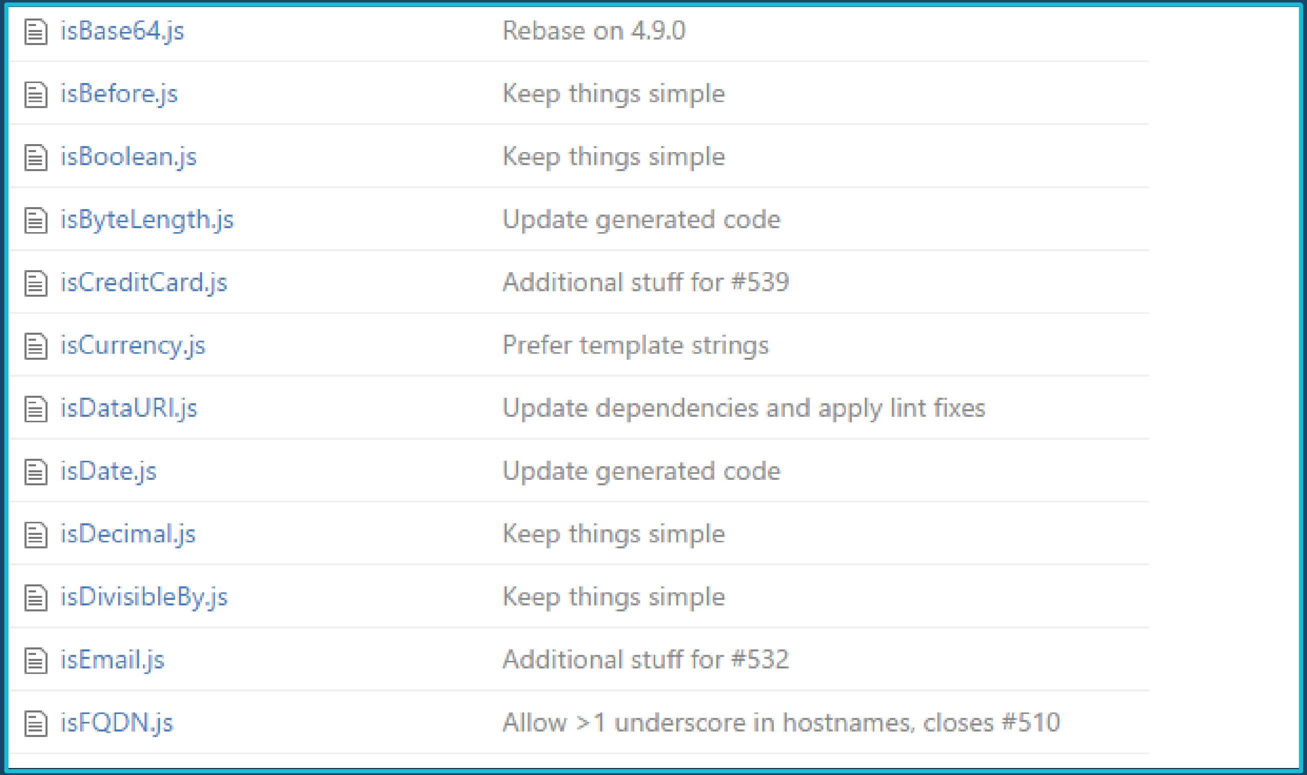
Regular Expression Denial of Service

Affected package	Vulnerable versions	Latest version
negotiator	<= 0.6.0	0.6.1

Best Practices

◇ Validator.js node.js module

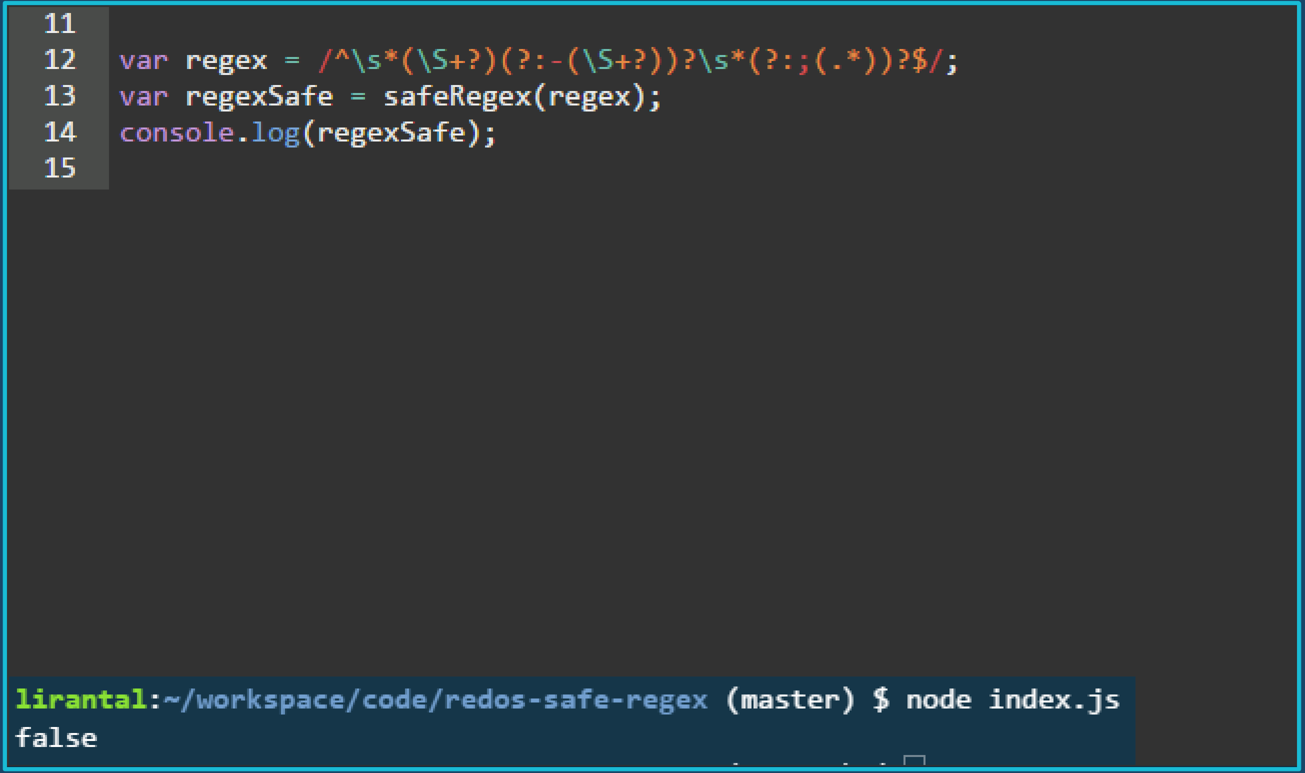




🗎 isBase64.js	Rebase on 4.9.0
🗎 isBefore.js	Keep things simple
🗎 isBoolean.js	Keep things simple
🗎 isByteLength.js	Update generated code
🗎 isCreditCard.js	Additional stuff for #539
🗎 isCurrency.js	Prefer template strings
🗎 isDataURI.js	Update dependencies and apply lint fixes
🗎 isDate.js	Update generated code
🗎 isDecimal.js	Keep things simple
🗎 isDivisibleBy.js	Keep things simple
🗎 isEmail.js	Additional stuff for #532
🗎 isFQDN.js	Allow >1 underscore in hostnames, closes #510

Best Practices

- ◇ safe-regex node.js module
- ◇ checks regex complexity/backtracking vulnerability



```
var regex = /^\s*(\S+?)(?:-(\S+?))?\s*(?:;(.*))?$/;
var regexSafe = safeRegex(regex);
console.log(regexSafe);
```

```
lirantal:~/workspace/code/redos-safe-regex (master) $ node index.js
false
```

Best Practices

- OWASP Validation RegEx Repo

5

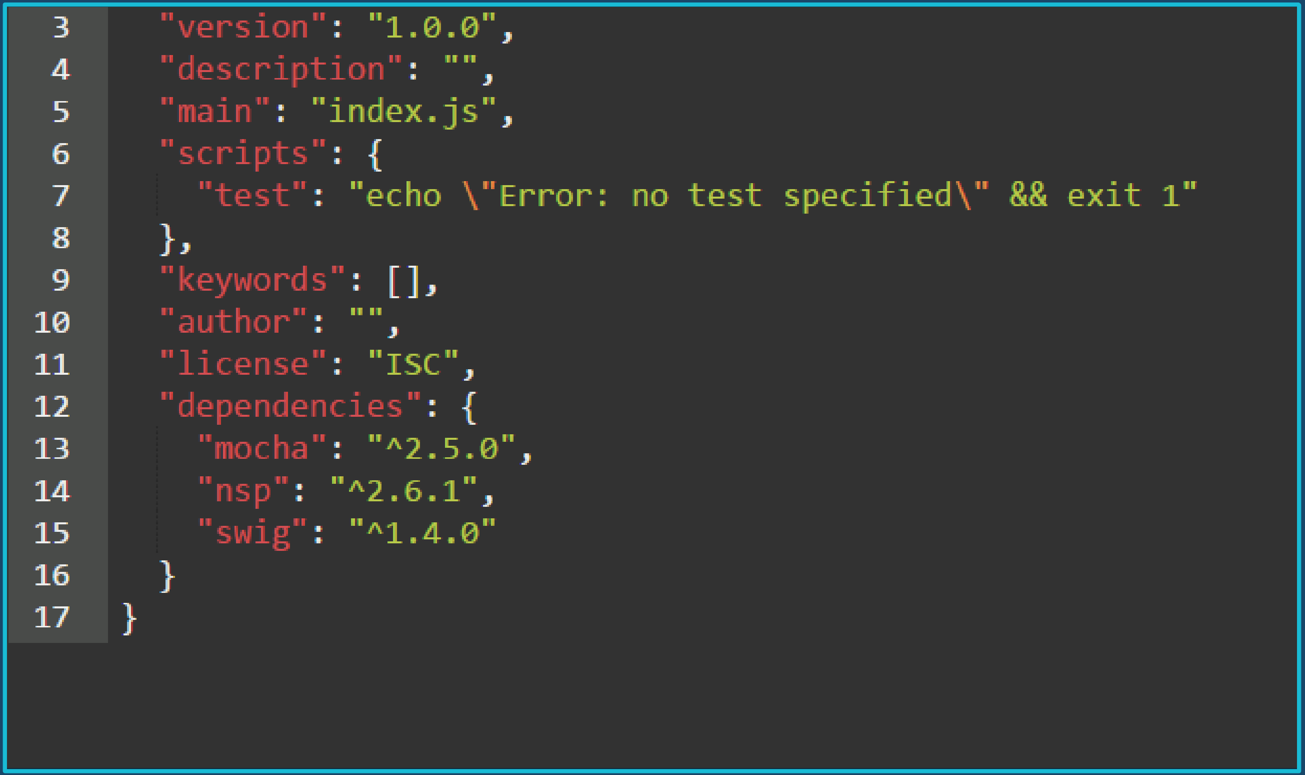
Vulnerability Scan

ask yourself

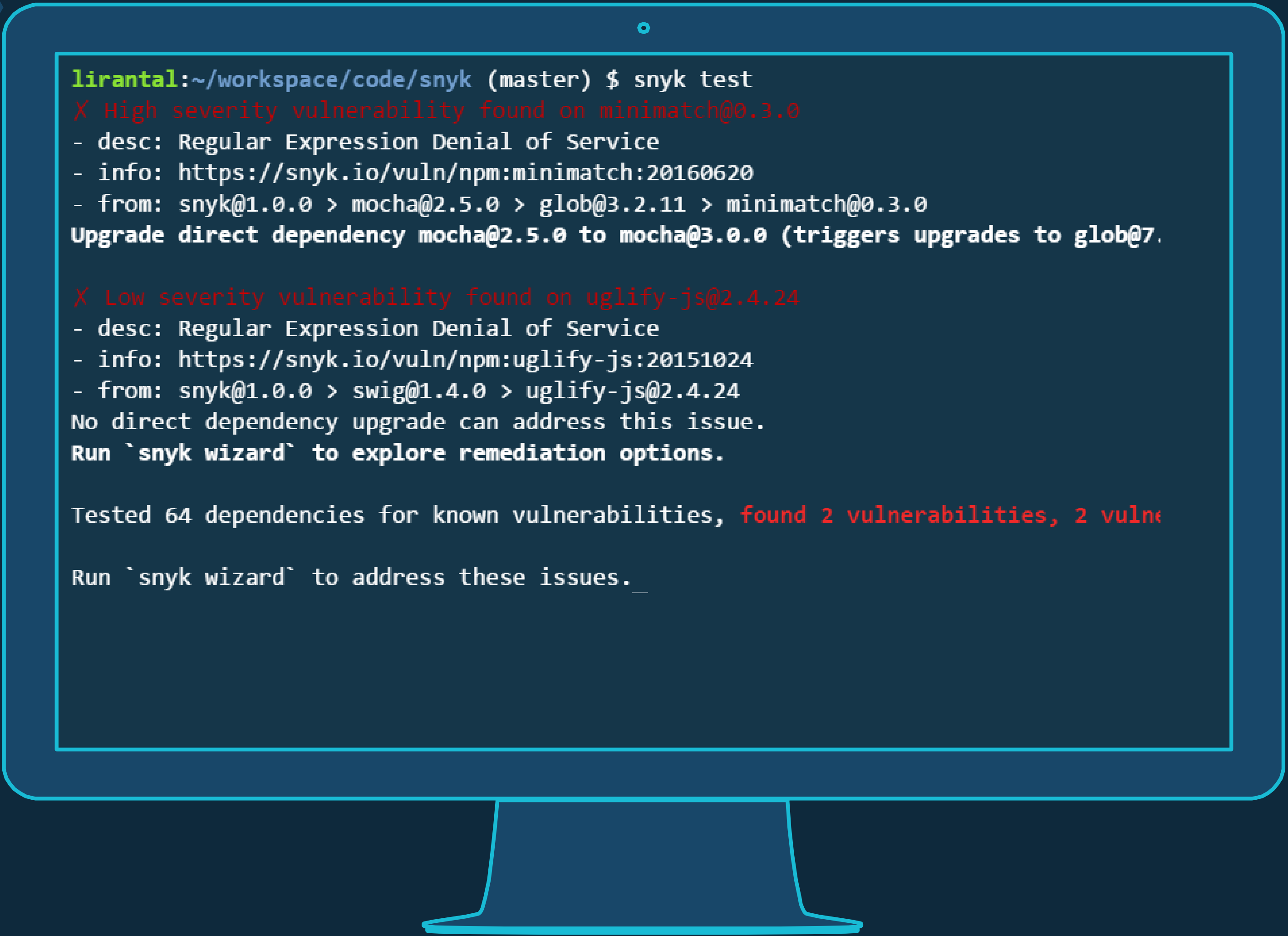
Are my dependencies vulnerable?

snyk

- check cve db for known issues
- check installed node_modules dir
- provides patch-level fix
- provides interactive patch wizard



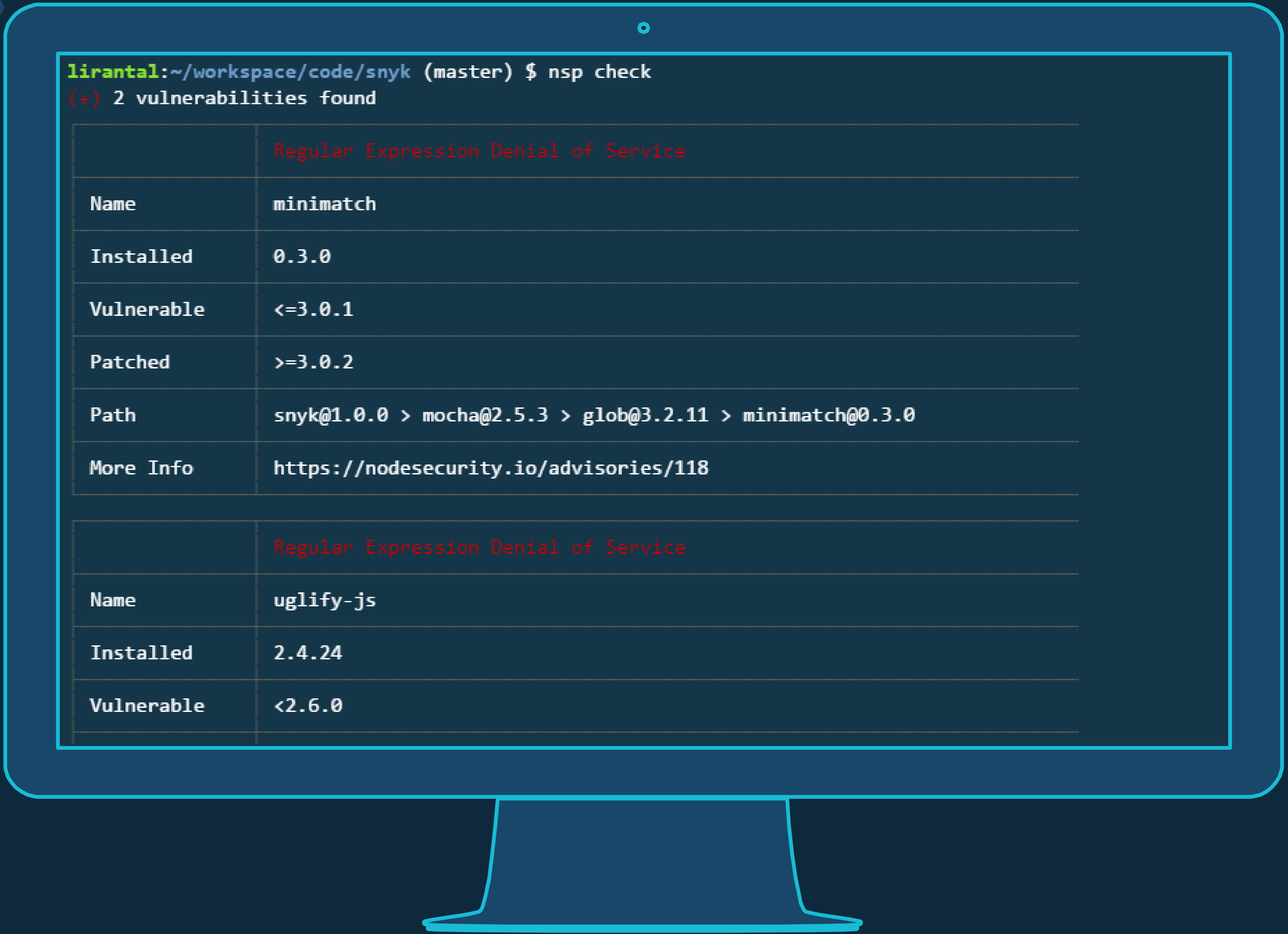
```
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "mocha": "^2.5.0",
    "nsp": "^2.6.1",
    "swig": "^1.4.0"
  }
}
```



```
lirantal:~/workspace/code/snyk (master) $ snyk test
✗ High severity vulnerability found on minimatch@0.3.0
- desc: Regular Expression Denial of Service
- info: https://snyk.io/vuln/npm:minimatch:20160620
- from: snyk@1.0.0 > mocha@2.5.0 > glob@3.2.11 > minimatch@0.3.0
Upgrade direct dependency mocha@2.5.0 to mocha@3.0.0 (triggers upgrades to glob@7.

✗ Low severity vulnerability found on uglify-js@2.4.24
- desc: Regular Expression Denial of Service
- info: https://snyk.io/vuln/npm:uglify-js:20151024
- from: snyk@1.0.0 > swig@1.4.0 > uglify-js@2.4.24
No direct dependency upgrade can address this issue.
Run `snyk wizard` to explore remediation options.

Tested 64 dependencies for known vulnerabilities, found 2 vulnerabilities, 2 vulne

Run `snyk wizard` to address these issues._
```

nsp

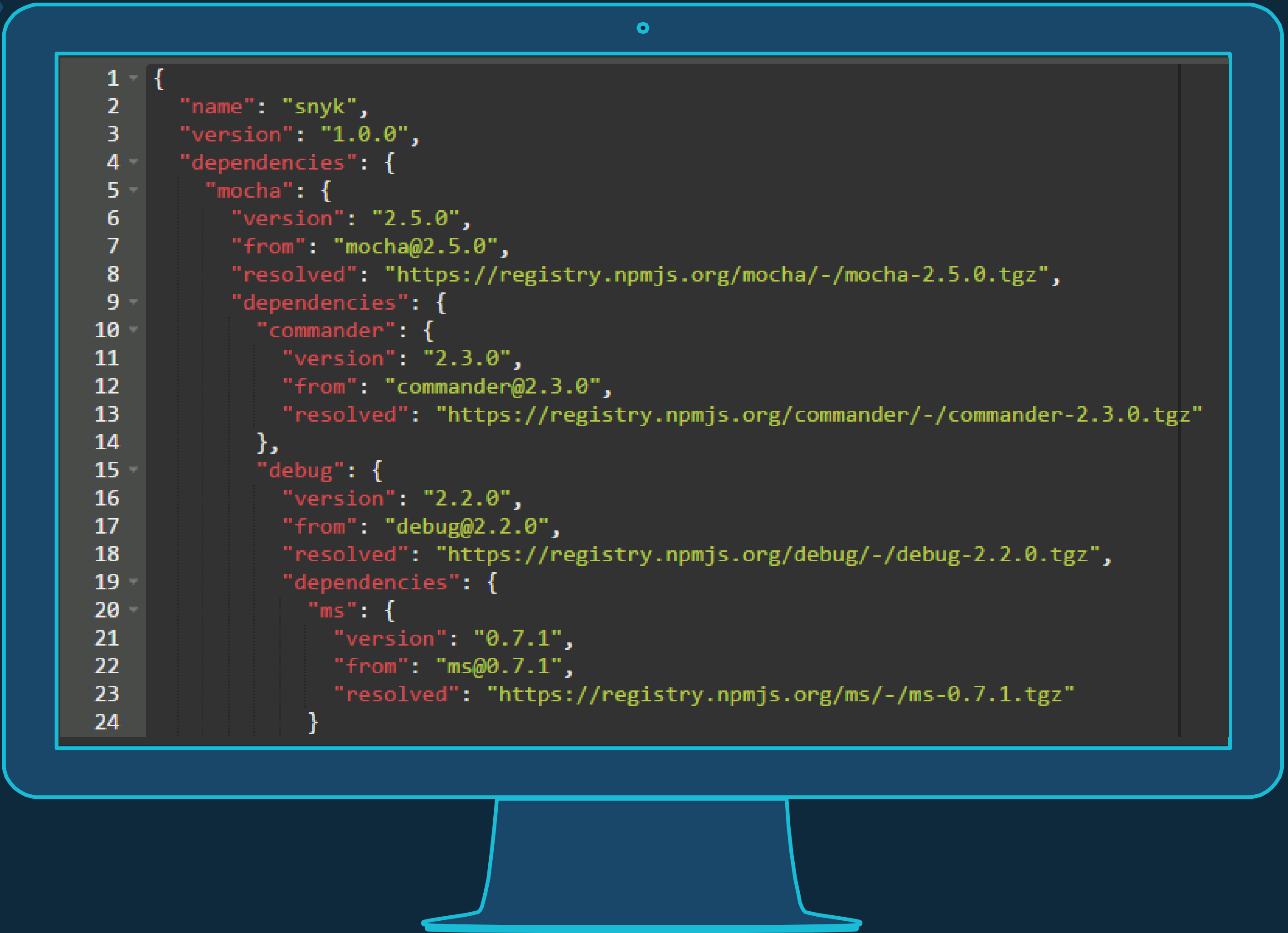
- check cve db for known issues
- check installed node_modules dir



```
lirantal:~/workspace/code/snyk (master) $ nsp check
(+) 2 vulnerabilities found

            Regular Expression Denial of Service

 Name       minimatch

 Installed  0.3.0

 Vulnerable <=3.0.1

 Patched    >=3.0.2

 Path       snyk@1.0.0 > mocha@2.5.3 > glob@3.2.11 > minimatch@0.3.0

 More Info  https://nodesecurity.io/advisories/118


            Regular Expression Denial of Service

 Name       uglify-js

 Installed  2.4.24

 Vulnerable <2.6.0
```

shrinkwrap

- pin-down dependencies
- pin-down devDependencies
- ship with tested packages
- avoid surprises in production build



```
{
  "name": "snyk",
  "version": "1.0.0",
  "dependencies": {
    "mocha": {
      "version": "2.5.0",
      "from": "mocha@2.5.0",
      "resolved": "https://registry.npmjs.org/mocha/-/mocha-2.5.0.tgz",
      "dependencies": {
        "commander": {
          "version": "2.3.0",
          "from": "commander@2.3.0",
          "resolved": "https://registry.npmjs.org/commander/-/commander-2.3.0.tgz"
        },
        "debug": {
          "version": "2.2.0",
          "from": "debug@2.2.0",
          "resolved": "https://registry.npmjs.org/debug/-/debug-2.2.0.tgz",
          "dependencies": {
            "ms": {
              "version": "0.7.1",
              "from": "ms@0.7.1",
              "resolved": "https://registry.npmjs.org/ms/-/ms-0.7.1.tgz"
            }
```

SecurityOps

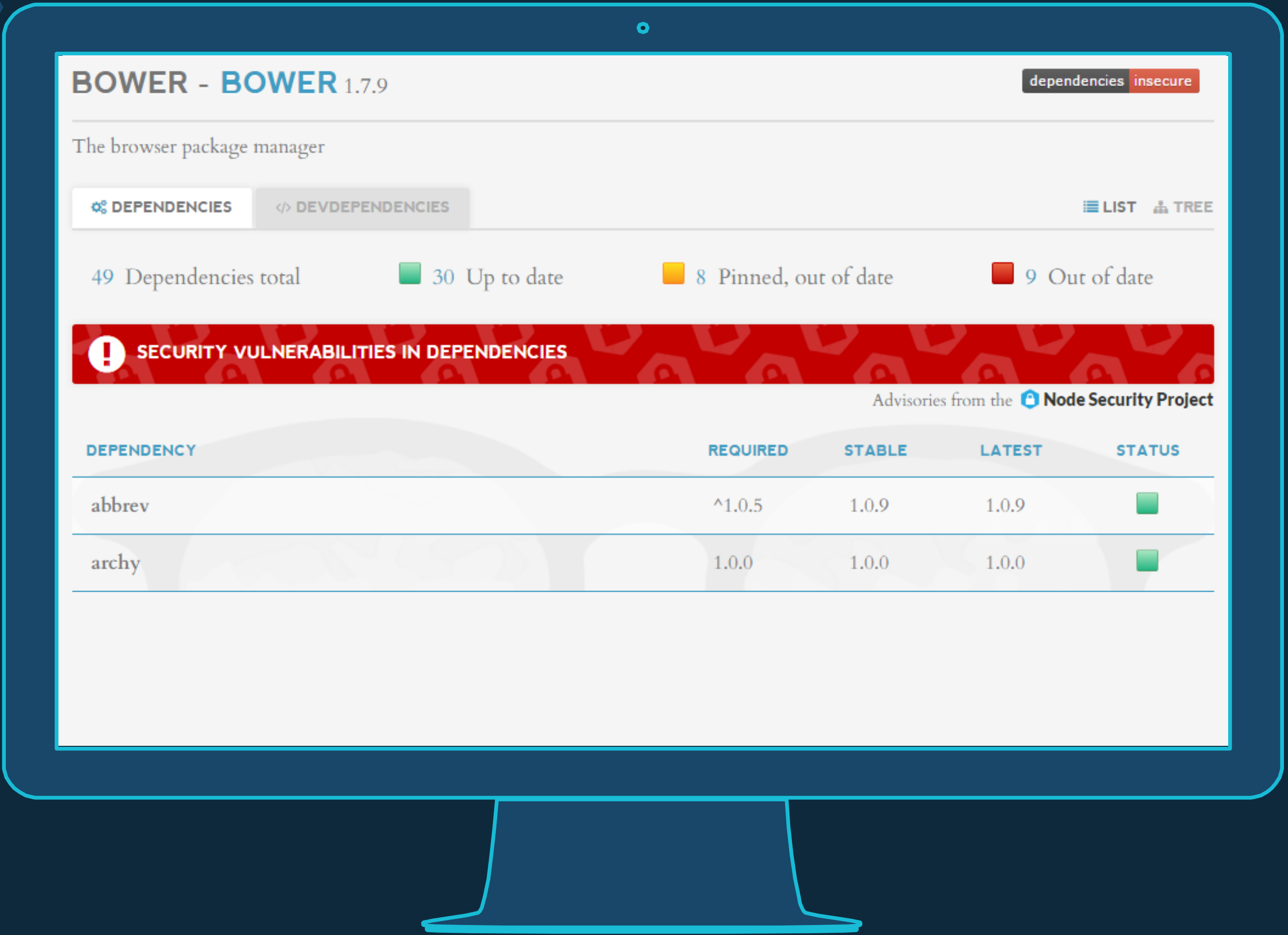
Integrated Security into your build pipeline

6

Security as a Service

david-dm

- monitor nodejs dependencies
- check installed node_modules dir



BOWER - BOWER 1.7.9

dependencies insecure

The browser package manager

DEPENDENCIES DEVDEPENDENCIES LIST TREE

49 Dependencies total 🟩 30 Up to date 🟧 8 Pinned, out of date 🟥 9 Out of date

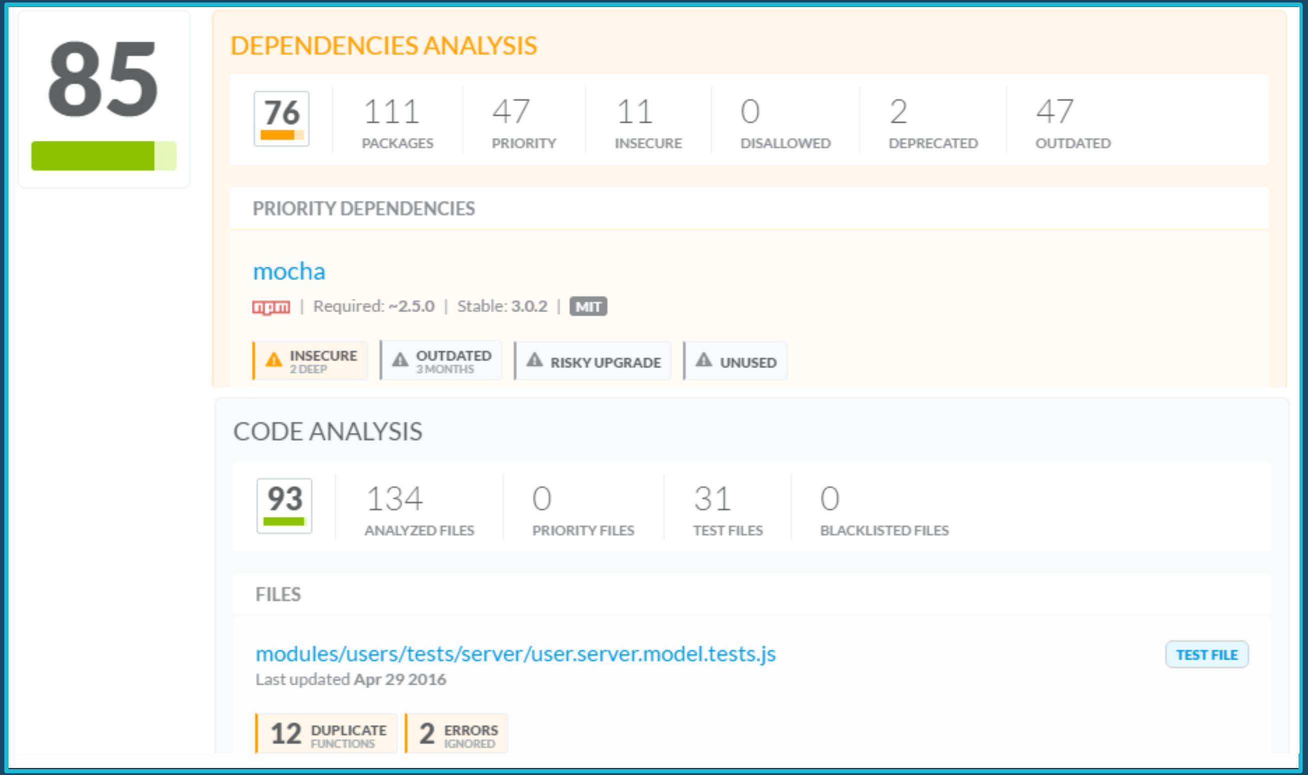
SECURITY VULNERABILITIES IN DEPENDENCIES

Advisories from the Node Security Project

DEPENDENCY	REQUIRED	STABLE	LATEST	STATUS
abbrev	^1.0.5	1.0.9	1.0.9	🟩
archy	1.0.0	1.0.0	1.0.0	🟩

Bithound.io

- monitor nodejs dependencies
- lint / static code analysis



85

DEPENDENCIES ANALYSIS

76

111 PACKAGES

47 PRIORITY

11 INSECURE

0 DISALLOWED

2 DEPRECATED

47 OUTDATED

PRIORITY DEPENDENCIES

mocha

npm | Required: ~2.5.0 | Stable: 3.0.2 | MIT

INSECURE 2 DEEP | OUTDATED 3 MONTHS | RISKY UPGRADE | UNUSED

CODE ANALYSIS

93

134 ANALYZED FILES

0 PRIORITY FILES

31 TEST FILES

0 BLACKLISTED FILES

FILES

modules/users/tests/server/user.server.model.tests.js TEST FILE

Last updated Apr 29 2016

12 DUPLICATE FUNCTIONS | 2 ERRORS IGNORED

Summary:

1. Helmet or Lusca for secure HTTP headers
2. Obsecure the session name
3. Validate and Sanitize req.body params to NoSQL

Summary:

4 Use validator.js for regex

5 Dependencies check with snyk, and nsp

6 SaaS Security with bithound.io and david-dm

Thanks!

Any questions?

- liran.tal@hpe.com
- GitHub