
OWASP
AppSec EU
Belfast
8th to 12th
of May
2017
Waterfront
Conference
Center


# Node.js

## Could a few lines of code F@#k it all up?

Erez Yalon
Head of AppSec Research
CHECKMARX

erez.yalon@checkmarx.com
@ErezYalon



# Could a few lines of code F@#k it all up?

Short answer: **YES!**

Longer answer: **Definitely YES!**

# What if the vulnerable line is this?

```
var x = require('nodepackage');
```

# Open Source

- 65%-90% of commercial application make use of Open Source Software
- Open Source is great but….
- Node.js is a leading framework
- Some issues with node’s repository are concerning from the security point of view

# The left-pad fiasco (March 2016)

## Azer Koçulu

A fairly anonymous developer that decided to “Liberate” his Node.js packages following a disagreement with npm staff

Among his modules was a little module named ***left-pad***

```
1  module.exports = leftpad;
2  function leftpad (str, len, ch) {
3    str = String(str);
4    var i = -1;
5    if (!ch && ch !== 0) ch = ' ';
6    len = len - str.length;
7    while (++i < len) {
8      str = ch + str;
9    }
10   return str;
11 }
```

# The left-pad fiasco

**silkentrance** commented on Mar 22

When building projects on travis, or when searching for left-pad on npmjs.com, both will report that the package cannot be found.

Here is an excerpt from the travis build log

```
npm ERR! Linux 3.13.0-40-generic
npm ERR! argv "/home/travis/.nvm/versions/node/v4.2.2/bin/node" "/home/travis/.nvm/versions/node/v4
npm ERR! node v4.2.2
npm ERR! npm  v2.14.7
npm ERR! code E404
npm ERR! 404
npm ERR! 404 'left-pad' is not in the npm registry.
npm ERR! 404 You should bug the author to publish it (or use the name yourself!)
npm ERR! 404 tarball, folder, http url, or git url.
npm ERR! Please include the following file with any support request:
npm ERR!     /home/travis/build/coldrye-es/pingo/npm-debug.log
make: *** [deps] Error 1
```

npm ERR! 404 Registry returned 404 for GET on https://registry.npmjs.org/left-pad

npm ERR! 404 'left-pad' is not in the npm registry.
npm ERR! 404 You should bug the author to publish it (or use the name yourself!)

# The left-pad fiasco



Left-pad was used by ~40 npm modules including React and Babel (used by FaceBook, AirBnB and others)

- First of all Azer is no longer anonymous.
- He actually triggered an important discussion within the community
- Should an author be able to un-publish his work without a process?
- What happens to the available module names?

# The npm platform

## Node.js Package Manager

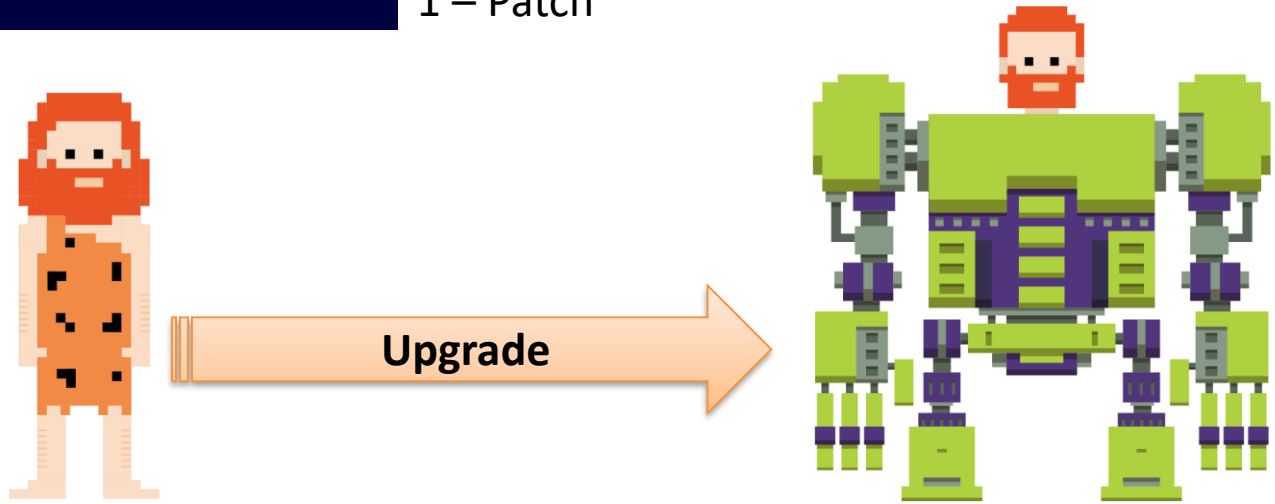
- Open source package manager
- “Find, share, and reuse packages of code from hundreds of thousands of developers”
- Around 450K modules available

# Some points to note about npm repo

- npm encourages the use of ***semver*** - **semantic versioning**.

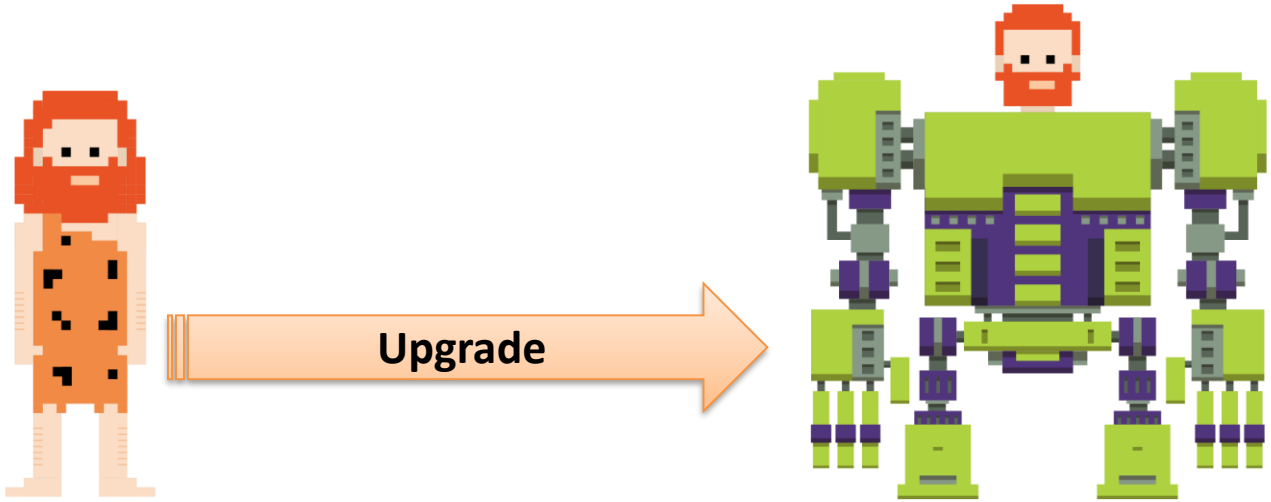
```
"dependencies": {
    "node-package": "^2.8.1"
}
```

2 – Major version
8 – Minor version
1 – Patch



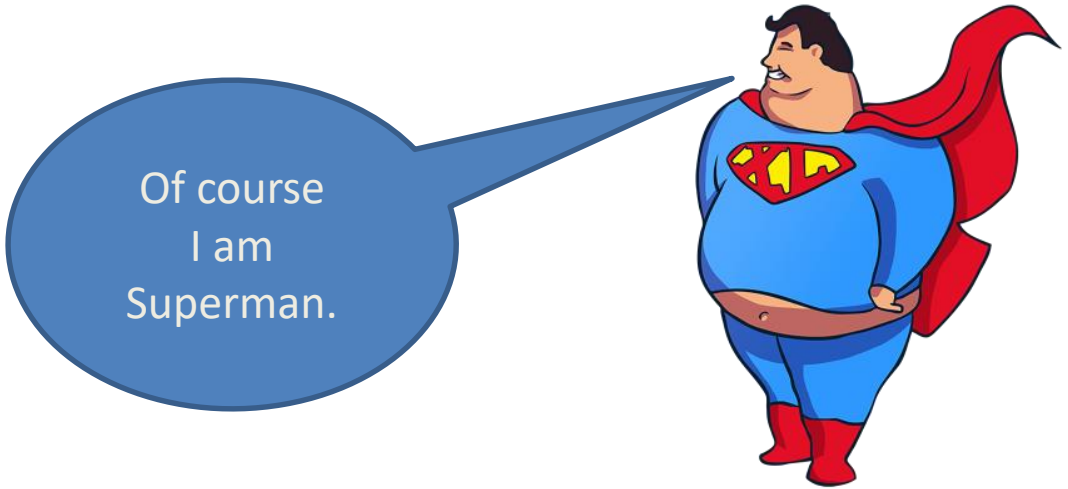
# Some points to note about npm repo

- npm encourages the use of ***semver*** - **semantic versioning**.
  - Dependencies are not locked to a certain version by default.
  - For any package, the author can push a new version at any time.



# Some points to note about npm repo

- npm utilizes **persistent authentication** to the npm server.
  - Users are not logged out until they voluntarily do so.



# Some points to note about npm repo

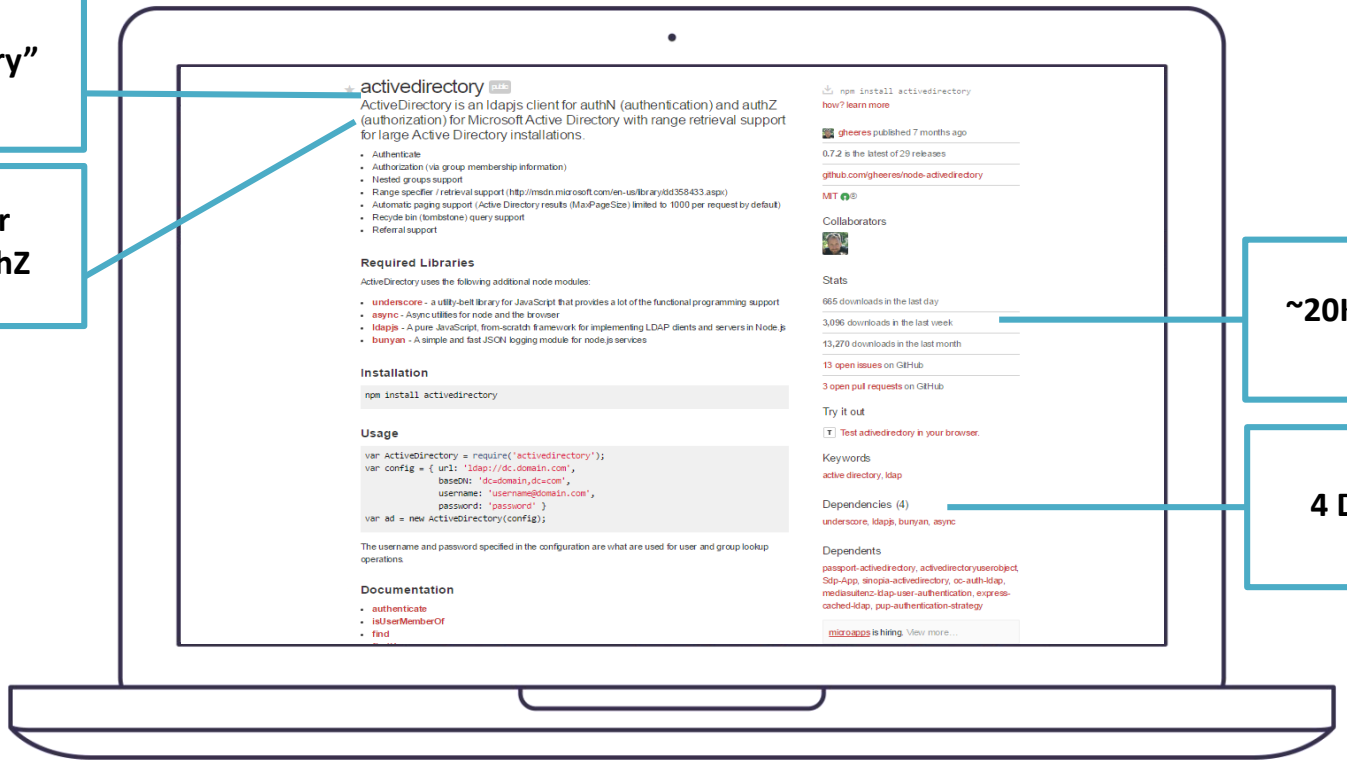
- Centralized registry – NPM utilizes a centralized registry
  - Typing ***npm publish*** ships your code to this registry server, where it can be installed by anyone.
    - **Any** user who is currently logged in and types ***npm install*** may allow any module to execute arbitrary publish commands

# Example time



**“activedirectory”**

**LDAP client for AuthN and AuthZ**




**~20K downloads last month**

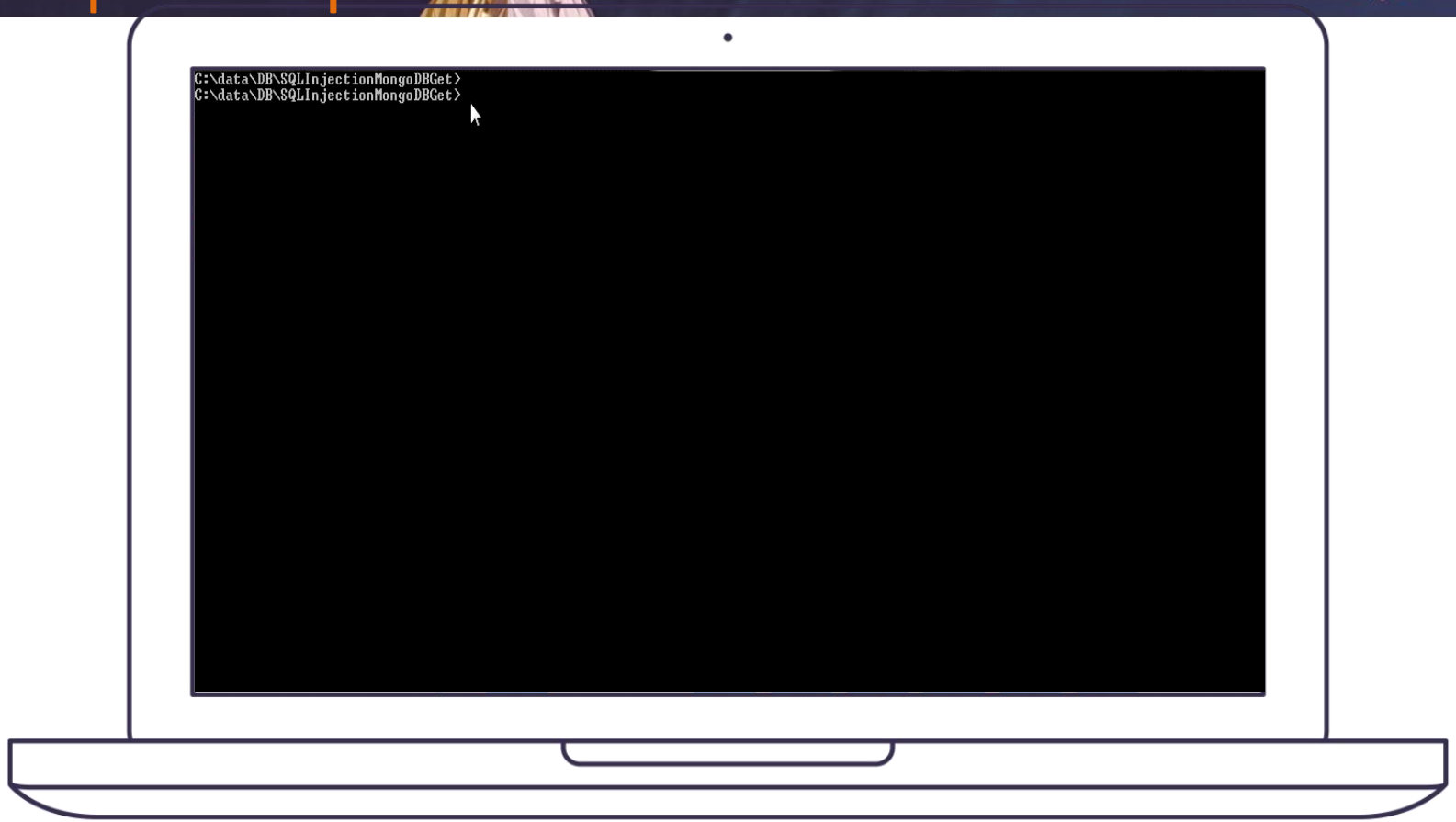
**4 Dependencies**



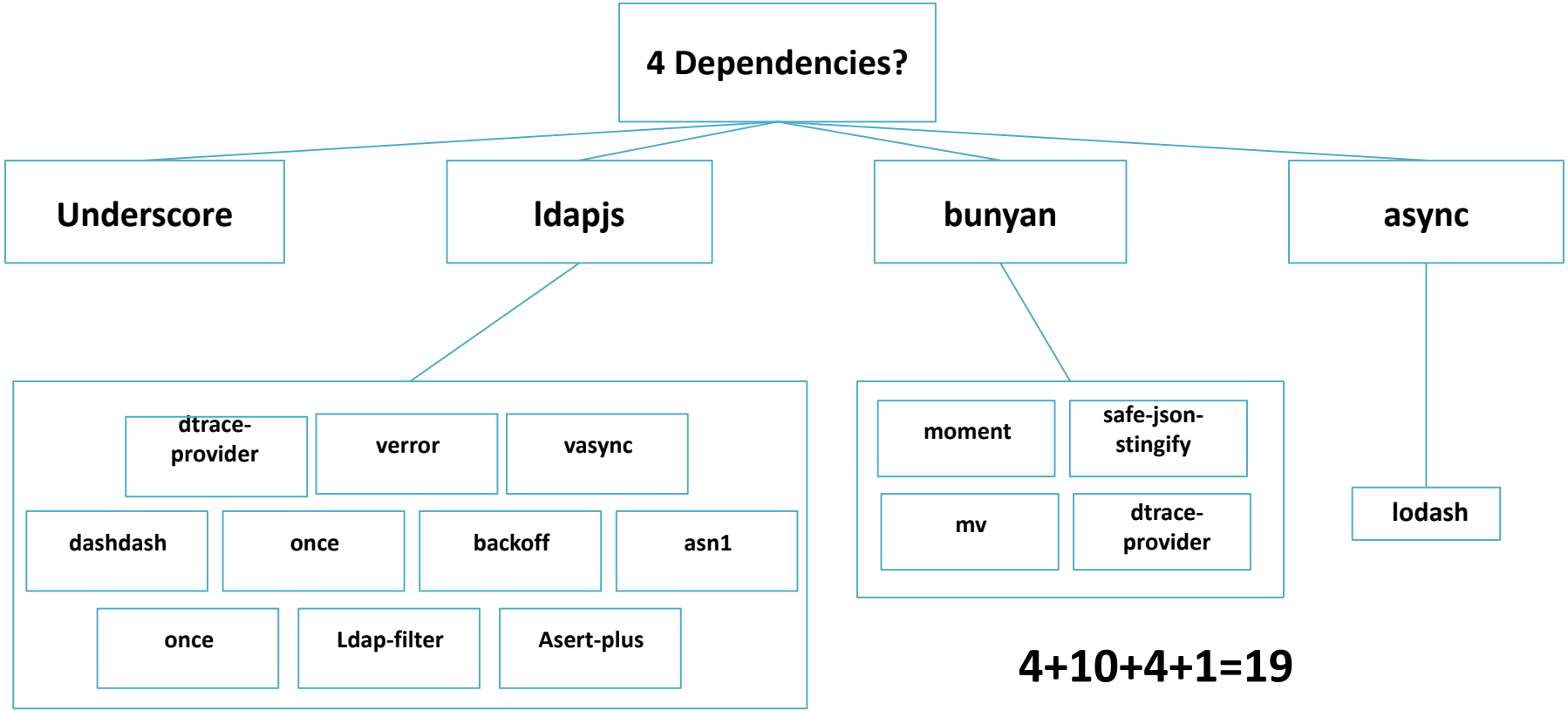
# 4 Dependencies?

## Let’s check

# So simple - npm install <module name>



# Lets take an example npm



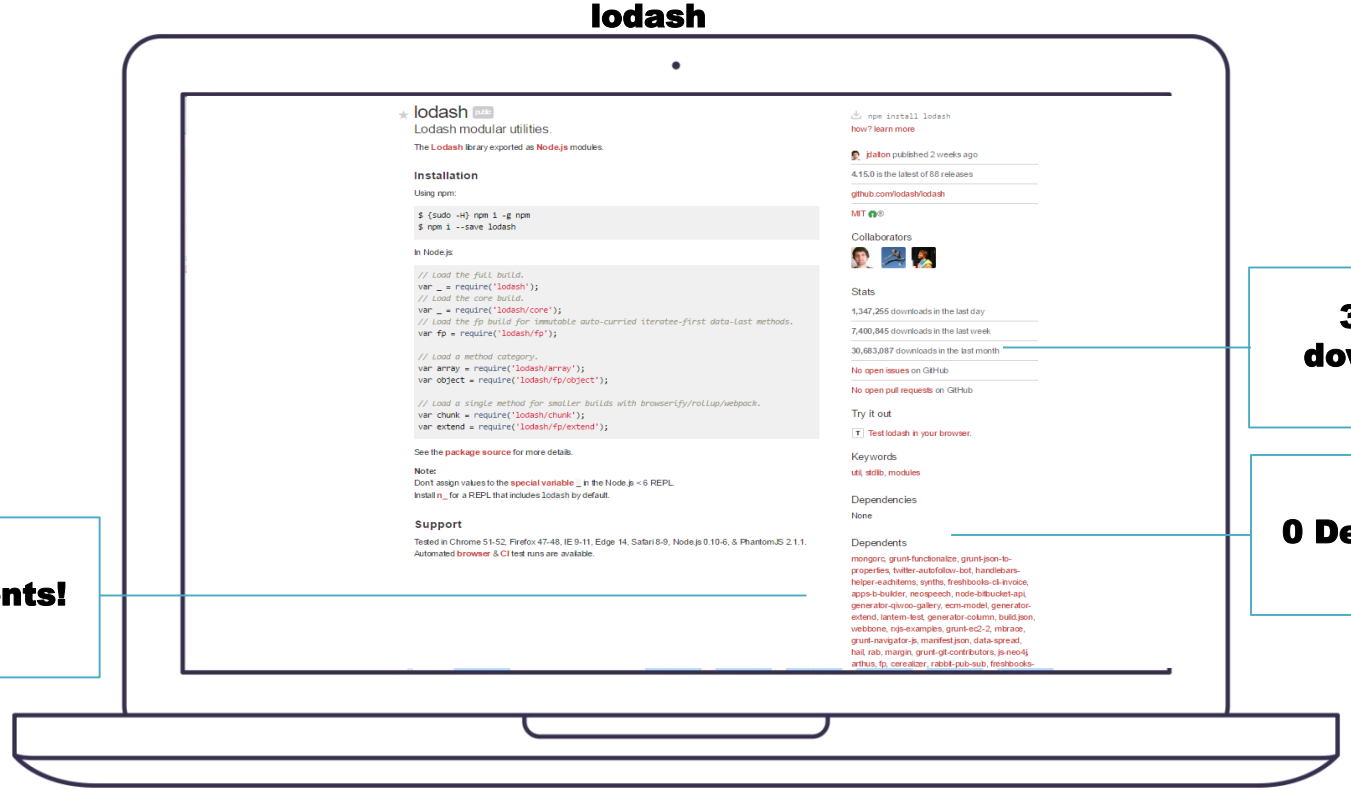
# Lets take an example npm



**lodash**

**lodash**

# What about lodash?




**45K Dependents!**

**30,683,087 downloads last month!**

**0 Dependencies?**



# Let’s take a look at some potential scenarios

# Ways to cause damage

- Create a useful module
  - Some good old marketing
  - Update it after it gets adoption
- Create module with similar name (Typo attacks)
- Taking over control of a legit account

  Packages are identified by names (No unique identifier/key)
- Create a self replicating worm

## npm hydra worm disclosure

Author: Sam Saccone

### tl;dr

It is possible for a single malicious npm package to spread itself across most of the npm ecosystem very quickly. This package could enable delivery of a potentially targeted, malicious payload to corporate entities.

**Full report by Sam Saccone:** https://www.kb.cert.org/CERT_WEB/services/vul-notes.nsf/6eacfaeab94596f5852569290066a50b/018dbb99def6980185257f820013f175/$FILE/npmwormdisclosure.pdf

# Creating a self replicating NPM worm (Lifecycle Scripts)

- Socially engineer an npm module owner to ***npm install*** an infected module
- Using installation scripts, the worm creates a new npm module

John

npm install
Hydra_A

```
"scripts": {
  "start": "node create malicious_npm_module",
  "predeploy": "echo im about to deploy",
  "postdeploy": "echo ive deployed",
  "prepublish": "coffee --bare --compile --output
lib/foo src/foo/*.coffee"
```

# Creating a self replicating NPM worm (Persistent Authentication)

- Worm sets lifecycle hook on the new module to execute the worm on install
- Worm publishes the new module to the user's npm account

John

npm publish

John
Legit 1
Legit 2
**malicious_npm_module**

# Creating a self replicating NPM worm (Semantic Versioning)

- Worm traverse through all user's npm modules (publish permissions) and adds the new malicious module as a dependency in their package.json.
- Worm publishes new versions to each of the modules with a version bump at the patch level semver – masked as "hotfix"

John

Package.json

```
"dependencies": {
  "primus": "*",
  "async": "~0.8.0",
  "express": "4.2.x",
  "malicious_npm_module": "
```

# From Malicious to Careless

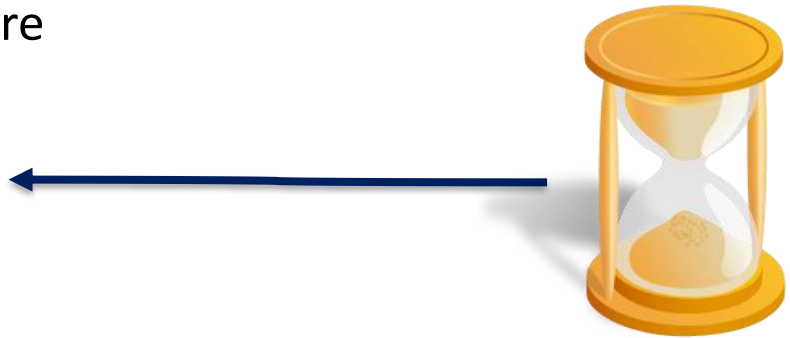
# What is wrong with this picture?

```
return function middleware (req, res, next) {

  // Strip any null bytes from the url
  while(req.url.indexOf('%00') !== -1) {
    req.url = req.url.replace(/\%00/g, '');
  }
```

# What we did

- Scan Node.js packages looking for vulnerabilties
  - Top 50 popular packages
  - Top 50 dependent-upon packages
  - Other popular packages
- Analyze results
- Responsible Disclosure
  - Contact dev
  - Wait for patch
  - Publish



# npm top 50



## Packages people 'npm install' a lot

**browserify**
browser-side require() the nod...
13.0.1 published 4 months ago by jmm

**grunt-cli**
The grunt command line interf...
1.2.0 published 5 months ago by vladikoff

**bower**
The browser package manager
1.7.9 published 5 months ago by sheerun

**gulp**
The streaming build system
3.9.1 published 7 months ago by phated

**grunt**
The JavaScript Task Runner
1.0.1 published 5 months ago by shama

express

**express**
Fast, unopinionated, minimali...
4.14.0 published 3 months ago by dougwilson

**npm**
a package manager for JavaSc...
3.9.6 published 3 months ago by zkat

**cordova**
Cordova command line interfa...
6.2.0 published 4 months ago by stevegill

**forever**
A simple CLI tool for ensuring ...
0.15.2 published 4 months ago by indexzero

https://www.npmjs.com/

## Most depended-upon packages

**lodash**
Lodash modular utilities.
4.15.0 published 4 weeks ago by jdalton

**request**
Simplified HTTP request client.
2.74.0 published 2 months ago by simov

**async**
Higher-order functions and common patt...
2.0.1 published 2 months ago by megawac

**underscore**
JavaScript's functional programming help...
1.8.3 published a year ago by jashkenas

**express**
Fast, unopinionated, minimalist web fra...
4.14.0 published 3 months ago by dougwilson

**commander**
the complete solution for node.js comm...
2.9.0 published 11 months ago by zhiyelee

**bluebird**
Full featured Promises/A+ implementatio...
3.4.6 published a week ago by esailija

**chalk**
Terminal string styling done right. Much ...
1.1.3 published 5 months ago by qix

**debug**
small debugging utility
2.2.0 published a year ago by tootallnate

https://www.npmjs.com/

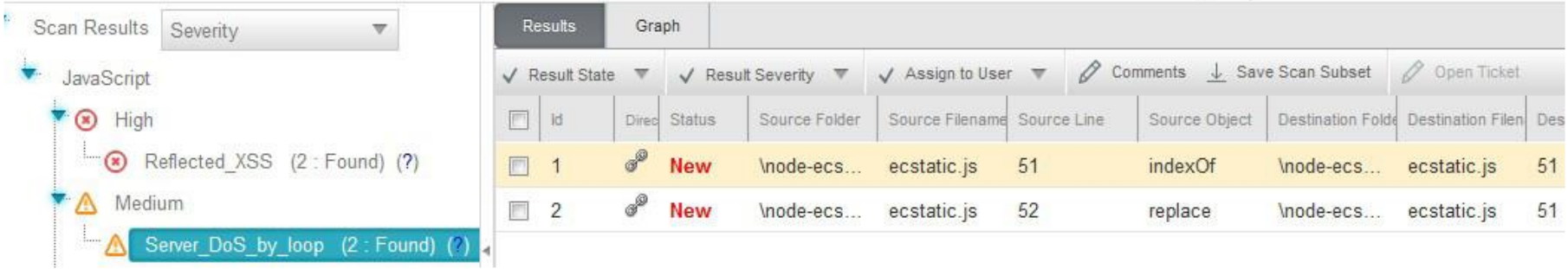
# Scan for security issues



| PROJECT NAME | LAST SCAN DATE | TEAM | LOC |
|---|---|---|---|
| acorn-master | 6/22/2016 | CxServer\SP\Company\npm project | 97564 |
| ansi-regex-master | 6/22/2016 | CxServer\SP\Company\npm project | 414 |
| esprima-master | 6/22/2016 | CxServer\SP\Company\npm project | 75907 |
| inherits-master | 6/23/2016 | CxServer\SP\Company\npm project | 47 |
| isarray-master | 6/23/2016 | CxServer\SP\Company\npm project | 25 |
| lodash-master | 6/23/2016 | CxServer\SP\Company\npm project | 128337 |
| object-keys-master | 6/23/2016 | CxServer\SP\Company\npm project | 419 |
| private-master | 6/23/2016 | CxServer\SP\Company\npm project | 198 |

```
46
47
48  return function middleware (req, res, next) {
49
50    // Strip any null bytes from the url
51    while(req.url.indexOf('%00') !== -1) {
52      req.url = req.url.replace(/\%00/g, '');
53    }
54    // Figure out the path for the file from the given url
55    var parsed = url.parse(req.url);
56    try {
57      decodeURIComponent(req.url); // check validity of url
58      var pathname = decodePathname(parsed.pathname);
59    }
60    catch (err) {
61      return status[400](res, next, { error: err });
62    }
63
64    var file = path.normalize(
65          path.join(root,
66            path.relative(
67              path.join('/', baseDir),
68              pathname
```

# What is wrong with this picture?



# What is wrong with this picture?

```
return function middleware (req, res, next) {

  // Strip any null bytes from the url
  while(req.url.indexOf('%00') !== -1) {
    req.url = req.url.replace(/\%00/g, '');
  }
```

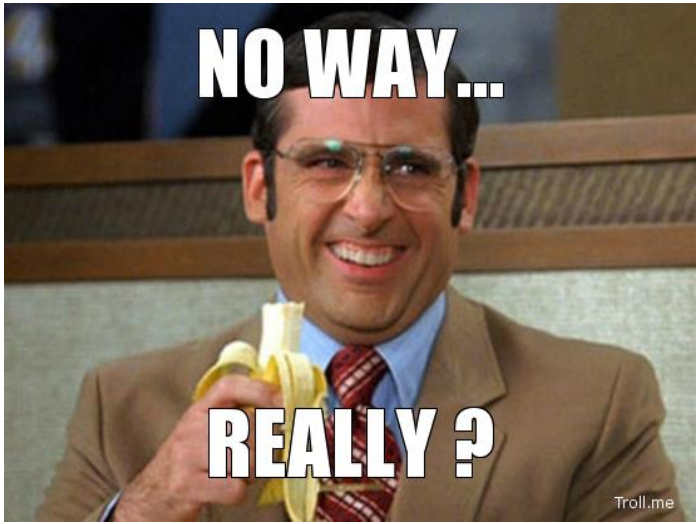
```
return function middleware (req, res, next) {

  // Strip any null bytes from the url
  while(req.url.indexOf('%00') !== -1) {
    req.url = req.url.replace(/\%00/g, '');
  }
```

**node-ecstatic**

- **~300K** downloads a month
- **~370** other npm packages are dependent on ecstatic

# Developer Response

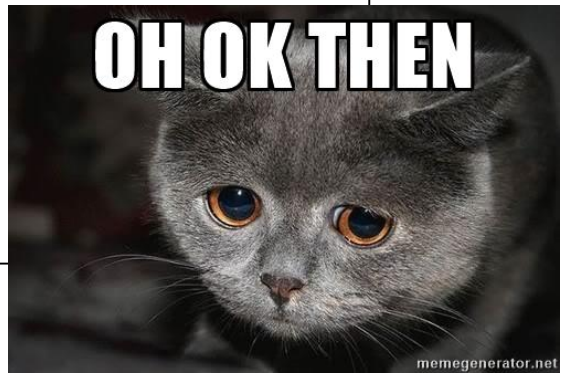



- PoC:
  - 22Kb payload - 1 sec lag
  - 35Kb payload - 3 sec lag
  - 86Kb payload - **server crashed**

http://www.checkmarx.com/%00%00%00%00%00%00 (...)

# Developer Response

```
return function middleware (req, res, next) {
  // Strip any null bytes from the url
  // This was at one point necessary because of an old bug in url.parse
  //
  // See: https://github.com/jfhbrook/node-ecstatic/issues/16#issuecomment-3039914
  // See: https://github.com/jfhbrook/node-ecstatic/commit/43f7e72a31524f88f47e367c3cc3af710e67c9f4
  //
  // But this opens up a regex dos attack vector! D:
  //
  // Based on some research (ie asking #node-dev if this is still an issue),
  // it's *probably* not an issue. :)
  /*
  while(req.url.indexOf('%00') !== -1) {
    req.url = req.url.replace(/\%00/g, '');
  }
  */
```



# Other Scan Results

- **Command Injection**
  - Variable from user input was used as an argument for an OS command.
  - Dev response:
    *“The flaw exists because the original author used it…*
    *A possible solution is to delete the vulnerable file”.*

# Other Scan Results

- **Command Injection**
- **Stored XSS**
- **Denial of Service** by Loop
- **Denial of Service** by Regex (**ReDoS**)
- **CSV Injection**
- **Insecure Randomness**
- **Open Redirect**



# So how do we protect ourselves?

# Be a Safe User!

- Inspect the code - http://registry.npmjs.org/MODULENAME/-/MODULENAME-VERSION.tgz
- Check if there are any hooks: ***npm show $module scripts***
- Don’t allow scripts to execute automatically: ***npm install –ignorescripts***
- Use ***npm shrinkwrap*** to lock down your own dependencies
- Sometimes it’s better to write your own functions!
- Analyze your code but your code includes your dependencies!
- Log out!

# Be a Safe Corporate!

- Run a local NPM repo
- Prevent installing from main registry

# Thank You.

Erez Yalon
Head of AppSec Research
CHECKMARX

erez.yalon@checkmarx.com
@ErezYalon