

# Check Yourself Before You Wreck Yourself

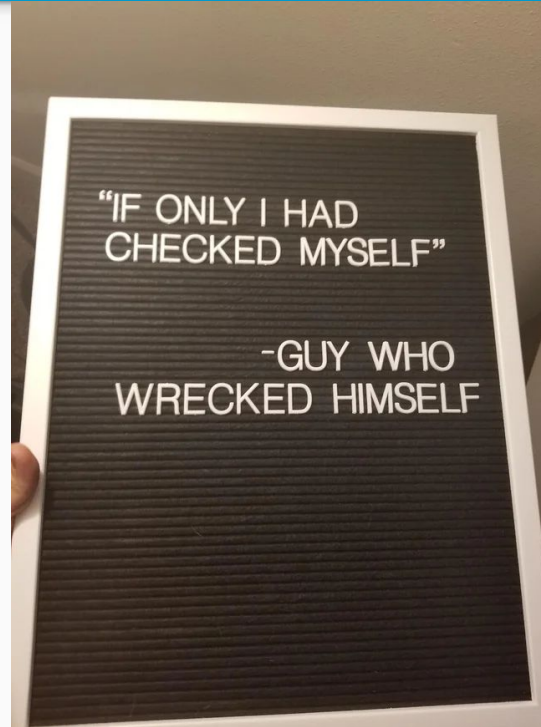
Auditing and Improving the Performance of Boomerang

Nic Jansma  
njansma@akamai.com  
@nicj



# Why are we here today?

- Boomerang: an open-source Real User Monitoring (RUM) third-party library
  - <https://github.com/akamai/boomerang>
- Why performance matters to us
- Performance Audit
- Improvements!
- Testing, Validation, Protecting against Regressions



# Why should you care?

- Do you develop a library that other teams, companies or projects use?
- Do you use a third-party library?
  - Any library that *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**.

Boss: Developer, please add this fancy new script!

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

What could go wrong? 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

# Boomerang

- 14,000+ mPulse sites
  - > 1 billion page loads a day
  
- 76,000 - 460,000 sites using open-source boomerang.js (estimate)

<https://discuss.httparchive.org/t/who-are-the-top-rum-analytics-providers/>

<https://trends.builtwith.com/javascript/Boomerang>

# 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**.

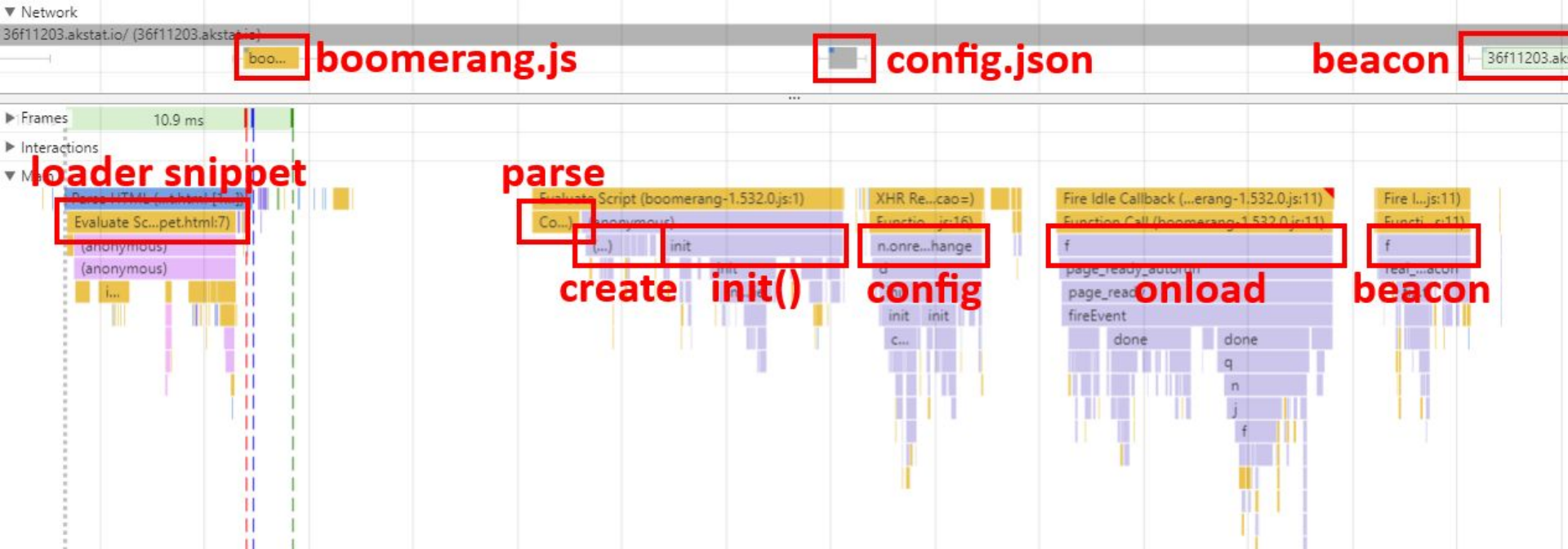
... but while it's the **easiest** way to judge a third party, it's just one aspect of the overall **cost**.

# A 3rd-Party Script's Lifecycle & Costs

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

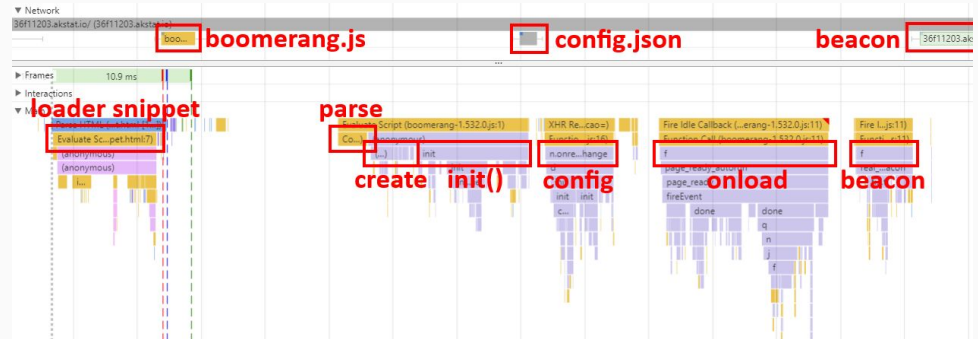


# Boomerang Performance Audit



# Boomerang Performance Audit

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



# A 3rd-Party Script's Lifecycle

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

Critical path!

1. Script **tag** itself has **no cost**: `<script src="..."></script>`
2. **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>
```

# Boomerang's Loader Snippet

1. **Loader Snippet / <script>**

2. Download

3. Parse + Compile

4. Initialize

5. Runtime / event handlers

3. **Boomerang's Loader Snippet**

Completely async and **non-blocking**

Better than `<script async>`

Cost: 2-40ms

**More expensive** than `<script>`, but guaranteed to not block

<https://akamai.github.io/boomerang/tutorial-loader-snippet.html>

# 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?

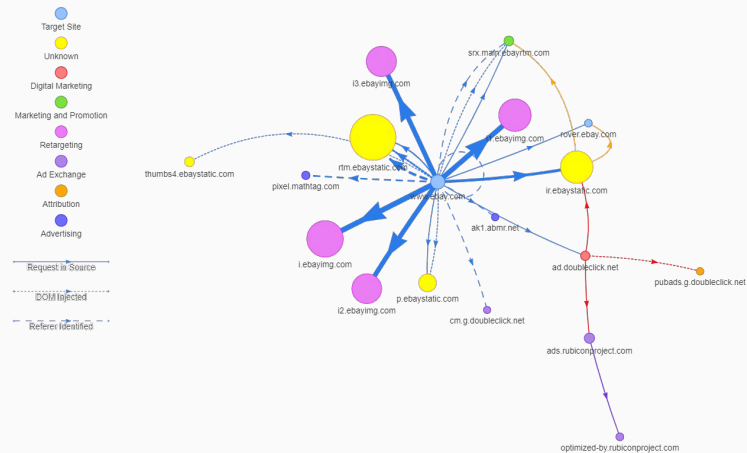
- External `<script>` / tag: **no (unless sharing domain)**
- Bundled with other components: **yes?**

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](http://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** **47 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

Boomerang is built with a plug-in architecture and **you** can build smaller builds if you'd prefer.

For example, if you don't need: SPA, XHR, UserTiming or Error Tracking support Boomerang **shrinks from 47 KB to 26 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
- Charting library responding to user interactions

# A 3rd-Party Script's Lifecycle

1. Loader Snippet / `<script>`

2. Download

3. Parse + Compile

4. Initialize

5. **Runtime / event handlers**

Boomerang: depending on the site, **10-40ms** at onload

Upwards of **300ms** on resource-heavy sites on low-end devices

# A 3rd-Party Script's Lifecycle

1. **Loader Snippet / <script>**

2. Download

3. **Parse + Compile**

4. Initialize

5. **Runtime / event handlers**

Critical path!

All **bold** could be done on the **main thread** (depending on the browser) and can cause **Long Tasks**.

# 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**.

# Boomerang's Performance Audit

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

TL;DR boomerang's 2018 **cost** (high-end to low-end devices):

1. Loader Snippet 2 - 40 ms
2. Download 188 KB raw / 47 KB gzip (non-blocking)
3. Parse 6 - 47 ms
4. Initialize 3 - 15 ms
5. @onload 10 - 300 ms
6. Beacon 2 - 20 KB
7. Runtime minimal

Tracking **improvements** @ <https://github.com/akamai/boomerang/issues>

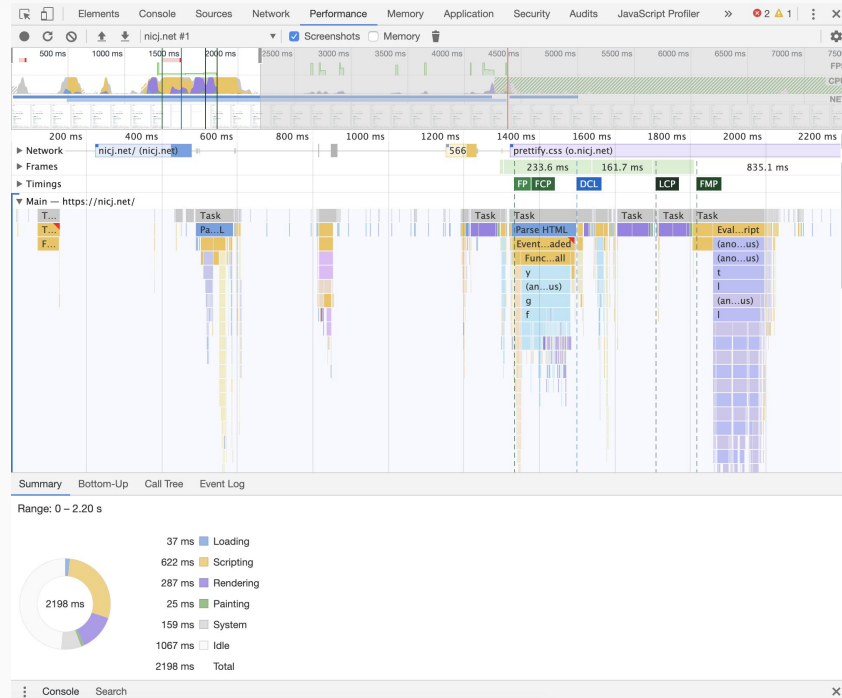
# Performance Audit Tools

Developer tools are your friend!

**Profilers** can point to **opportunities**

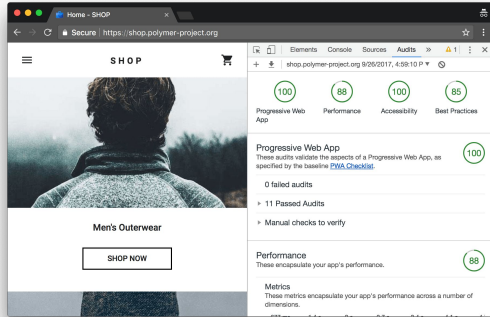
My advice:

- **Take your time**
- Get a sense for the **overall** picture
- **Look for extremes** - longest duration, tallest stack





# Evaluating for Performance

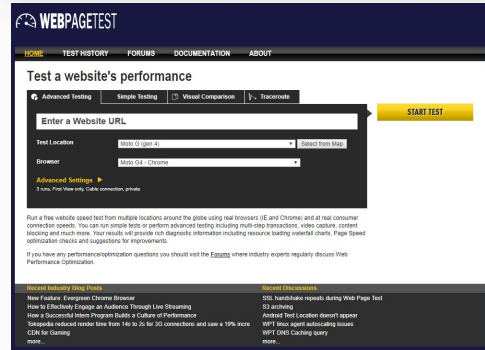
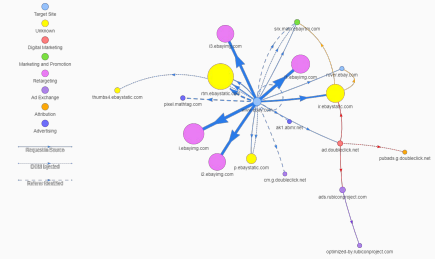


Chrome Lighthouse

[developers.google.com/web/tools/lighthouse/](https://developers.google.com/web/tools/lighthouse/)

RequestMap

[requestmap.webperf.tools](https://requestmap.webperf.tools)



WebPagemeter

[webpagemeter.org](https://webpagemeter.org)

3rdParty.io

[3rdparty.io](https://3rdparty.io)



# Boomerang's Performance Audit

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

We found room for improvement! Filed 15 issues. Examples:

- ResourceTiming Compression is expensive
- Loader Snippet Performance in Edge
- Breakup plugin creation / initialization to avoid long tasks
- Beacon: Review cookie access
- Beacon: Memory: Node counting is expensive
- Unload beacon size
- Unload Beacon: Memory plugin updating DOM counts

Tracking **improvements** @ <https://github.com/akamai/boomerang/issues>

# Boomerang's Performance Improvements

<https://nicj.net/boomerang-performance-update/>

- **New Loader Snippet**
- ResourceTiming Optimization
- Removed Debug Messages
- Improved Minification
- Reduced Cookie Size
- Reduced Cookie Access
- Simplified MD5 plugin
- Simplified SPA plugin
- Enabled Brotli for CDN

Using `<link rel="preload">` we can load **async** and **non-blocking** without an IFRAME

Reduced **2-40ms** to **1ms** for browsers that support Preload!

# Boomerang's Performance Improvements

<https://nicj.net/boomerang-performance-update/>

- New Loader Snippet
- **ResourceTiming Optimization**
- Removed Debug Messages
- Improved Minification
- Enabled Brotli for CDN
- Reduced Cookie Size
- Reduced Cookie Access
- Simplified MD5 plugin
- Simplified SPA plugin

Compressing ResourceTiming data was our **most expensive** task

Tweaked the algorithm slightly to be **slightly-less-than-perfect** for a **4x speedup**

Reduced some sites' cost from **100ms to 25ms** or **300ms to 75ms**

# Boomerang's Performance Improvements

<https://nicj.net/boomerang-performance-update/>

- New Loader Snippet
- ResourceTiming Optimization
- **Removed Debug Messages**
- **Improved Minification**
- **Enabled Brotli for CDN**
- Reduced Cookie Size
- Reduced Cookie Access
- Simplified MD5 plugin
- Simplified SPA plugin

We were shipping debug **log messages** even though the debug log was disabled (6% saving)

Changed from **Uglify2** to **Uglify3** (1.3% saving)

Enabled **Brotli** on the Akamai CDN (11.2% saving)

**SPA** and **MD5** plugins refactored (2.8% saving)

# Boomerang's Performance Improvements

<https://nicj.net/boomerang-performance-update/>

- New Loader Snippet
- ResourceTiming Optimization
- Removed Debug Messages
- Improved Minification
- Enabled Brotli for CDN
- **Reduced Cookie Size**
- **Reduced Cookie Access**
- Simplified MD5 plugin
- Simplified SPA plugin

We set a cookie to track **sessions**

Changed how we stored some of the data (e.g. **hash** instead of a full URL, **Base36** instead of Base10 for numbers): 41% smaller

We were reading/writing constantly during startup -- **simplified** our operations from 21 reads and 8 writes down to 2 reads and 4 writes

# Boomerang's Performance Improvements

<https://nicj.net/boomerang-performance-update/>

- New Loader Snippet
- ResourceTiming Optimization
- Removed Debug Messages
- Improved Minification
- Enabled Brotli for CDN
- Reduced Cookie Size
- Reduced Cookie Access
- **Simplified MD5 plugin**
- **Simplified SPA plugin**

We were using **MD5** for **hashing** and **comparing** URLs quickly

This plugin took **8.1 KB** and could hash **35,397 URLs/sec**

We replaced with the **FNV** algorithm: **0.34 KB** and **113,532 URLs/sec**

**SPA** plugin was simplified and removed framework-specific support in favor of just monitoring the `window.History` object

# Boomerang's Performance Audit

<https://nicj.net/boomerang-performance-update/>

After fixes:

1. Loader Snippet ~~2-40 ms~~ **1-20 ms (1 ms in modern browsers)**
2. Download ~~188 KB raw / 47 KB gzip~~ **196 KB raw / 47 KB brotli**
3. Parse 6 - 47 ms **(same)**
4. Initialize 3 - 15 ms **(same)**
5. @onload ~~10-300 ms~~ **5-75 ms**
6. Beacon 2 - 20 KB **(same)**
7. Runtime minimal

Tracking **improvements** @ <https://github.com/akamai/boomerang/issues>



# Boomerang's Performance Audit

<https://nicj.net/boomerang-performance-update/>

## Opportunities!

1. Loader Snippet ~~2-40 ms~~ **1-20 ms (1 ms in modern browsers)**
2. Download ~~188 KB raw / 47 KB gzip~~ **196 KB raw / 47 KB brotli**
3. Parse **6 - 47 ms (same)**
4. Initialize **3 - 15 ms (same)**
5. @onload ~~10-300 ms~~ **5-75 ms**
6. Beacon 2 - 20 KB **(same)**
7. Runtime minimal

Tracking **improvements** @ <https://github.com/akamai/boomerang/issues>

# Continuous, Gradual Improvement

*In a mature product with a healthy process you're much more likely to see a 50% gain come in the form of many 5% gains compounding to get to your goal via sustained effort and quality control*

<https://docs.microsoft.com/en-us/archive/blogs/ricom/the-performance-war-win-it-5-at-a-time>

# Protecting Against Regressions

Boomerang Performance Lab / Test Suite

Simple set of scenarios & metrics we capture each build

Tracks:


- CPU time via headless Profiler
- Counts & Durations via UserTiming marks & measures
- Sizes of code & plugins

<https://akamai.github.io/boomerang/tutorial-perf-tests.html>

# Realtime Telemetry

You can capture your script's own **runtime stats**, **Long Tasks** and **JavaScript errors**

JavaScript **Self Profiling API**



Boomerang Errors ▾ Last 7 Days ▾ All Beacon Types ▾ All Countries ▾ All Regions ▾ All Browser Families ▾ All Browsers ▾ All OS Families ▾ All OS ▾ All Dev  
All App Error Messages ▾ All App Error Types ▾ All App Error Sources ▾ 1,687.0 ▾

## App Errors

# 469

Row	App Error Message	App Errors	App Errors Percent
1	TypeError: Cannot read pr...erty 'clientHeight' of null	172	36.67%
2	Incorrect SPA time calculation	159	33.90%
3	(No Value)	72	15.35%
4	TypeError: null is not an object	34	7.25%
5	The operation is insecure	10	2.13%
6	Maximum call stack size exceeded	8	1.71%
7	Error: Not enough storag...o complete this operation	4	0.85%
8	TypeError: Cannot read pr...erty 'clientWidth' of null	4	0.85%
9	TypeError: 'c.document.body' is null	2	0.43%
10	TypeError: Cannot convert...ocument.body' to object	2	0.43%
11	TypeError: Object expected	1	0.21%
12	TypeError: undefined is not a function	1	0.21%
T...al		469	100.00%

# What can you do?

Boss: Developer, please add this fancy new script!

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

- Perform a **light-weight audit**
- Do its **benefits outweigh** its costs?
- **Ask** if the library has published performance information
- Every third-party should have an **owner** or **“internal champion”**

# What 3rd Party Scripts Should be Doing...

## They should:

- Use a CDN
- Compress resources
- Set caching headers
- Set Timing-Allow-Origin
- Set ACAO
- 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

3rdParty.io Home About Third-Parties

Nic Jansma

## 3rdParty.io

Best Practices for Third-Party Scripts

3rdParty.io monitors third-party scripts and libraries, and checks that they're following best practices for performance, reliability and security

Want to run a check-up on a script before you include it on your site?  
Want to see how your script fares?

Check!

## Akamai mPulse

Note: Browser evaluations are measured by including this library into a minimal web page. In the real world, the third-party library may require additional configuration, activation, page construction, or specific scenarios where it performs work. You should take the evaluations below as the *lower bound* of what this third party will cost.

### At a Glance

Loader Snippet

10ms

Fetch

31kB  
118ms

Startup

7ms parse  
6ms init

Overall CPU

33ms

Overall Network

1 reqs

3rdParty.io Home About Third-Parties

Nic Jansma

## Products

3rdParty.io Demo	3	3	37
Akamai mPulse	15	53	
Bitdiddle Analytics	5	11	46
Facebook Pixel	5	10	47
Facebook SDK for JavaScript	10	7	45
Google +1 Button	5	10	47

✓ HTTP Compression Yes

Last Updated: 2018-06-12T20:39:26.191Z

### Details

#### What?

Whether or not the content was compressed.

#### Why?

Compression reduces the number of bytes needed to fetch the script, and has a direct effect on the overall Page Weight.

#### How?

Ensure the HTTP server supports `Content-Encoding: gzip, deflate` or `br`

#### Methodology

Inspection of the `Content-Encoding` HTTP response header to determine whether or not the content was compressed.

#### Raw Data

`content-encoding: gzip`

# Links

- <https://nicj.net/an-audit-of-boomerangs-performance/>
- <https://nicj.net/boomerang-performance-update/>
- <https://github.com/akamai/boomerang/issues>
- <https://3rdparty.io/>



# thanks!

[nicj.net/talks/](https://nicj.net/talks/)

Nic Jansma  
njansma@akamai.com  
nic@nicj.net  
@nicj

