TALA

TECHNICAL OVERVIEW

Slow websites lose customers and hurt conversions.

YOU DON'T HAVE TO SACRIFICE PERFORMANCE FOR SECURITY.

Whether you're a webmaster or a marketer, you know how important website performance is. Your users want a fast experience and if there's too much friction, they'll move on:

- **A leading ticketing retailer lost 600 ticket sales per day** due to 50m/s latency, costing $3.5m in negative revenue impact.
- **The BBC lost 10% of users** for every additional second their website took to load.
- **47% of users expect a website to load in two seconds**; 40% will abandon it altogether if it takes longer than three.
- **79% of users wouldn't return to a site** that had previously performed poorly for them.

The problem is, you also need to secure your website and protect your users against Magecart and other data theft – because if their credit card is skimmed via your site, they won't be coming back again either.

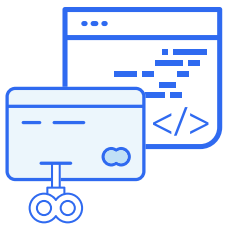
Running large numbers of third-party scripts and applications slows down website performance, but some web security solutions can have a similar impact. For example, JavaScript-based solutions faced with large numbers of APIs on a webpage can cause significant performance degradation, to the detriment of topline revenue:

Some solutions attempt to mask latency aspects such as Time to First Byte by working asynchronously or splitting the JavaScript into multiple files. The real issue, however, is what the JavaScript itself is doing in the browser and client machine; how long is it taking for content to render on the page and become responsive, enabling the user to genuinely interact.

> “When a page opens in a browser, large numbers of APIs are provided, often 200 or more. To be truly comprehensive and secure, a JavaScript-based solution has to hook into all of those APIs. *All of them.* If even one API is missed, the door is open.”

JavaScript runs even more slowly on mobile devices – no platform other than Apple is optimized for it. There are other metrics to consider too, including first interactive and first CPU idle...

Think about it: when a page in a browser, large numbers of APIs are provided, often 200 or more. To be truly comprehensive and secure, a JavaScript-based web security solution has to hook into all of those APIs. *All of them.* From a performance perspective, you're now running in synchronous blocking mode – nothing else runs or loads until this hooking is complete, effectively making this process a single point of failure for your website. And to cap it all, the solution itself is providing a hook for attack: if even one API is missed, the door is open. You can read more about the performance penalties of JavaScript insertion here.

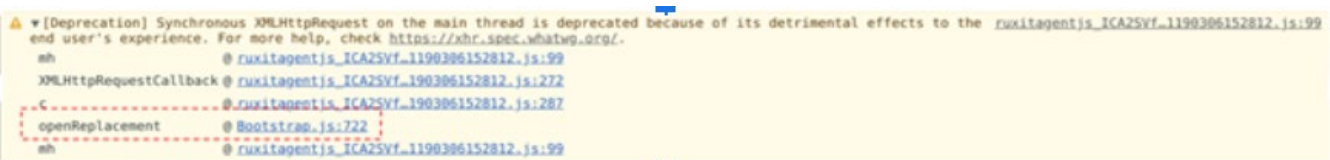


MEASURING UP

JavaScript virtualization solutions result in a measurable performance degradation of 100m/s to *over one second.* To measure how your JavaScript-based security impacts your web page loading performance, you can use Google Lighthouse Tool. The reasons behind the degraded performance you'll notice include:

- JS protection code must load first in a synchronous manner, i.e. your webpage's loading is blocked until the JS code has loaded first.
- Typically, JS-based solutions use Sync XHR to load policy configurations (and loading policy configuration after loading without adequate security controls, can itself be a security issue). Again, this block page rendering. Google is working on blocking sync XHRs, for all the good reasons, making this feature of your JS-based security solution problematic.

If you're using a JS-based security solution, you're likely to see the following error in Chrome's console tab:



(Note: this warning will be pointed to the JS file either directly or within the call stack).

TALA

IT DOESN'T HAVE TO BE THIS WAY

No one is going to abandon JavaScript and the rich web experience it enables any time soon. The good news is, you don't have to degrade your website's performance at all to secure it. The same experts who developed the powerful in-browser functionality we love to use also developed powerful tools to secure that experience. And because they're browser-native, they're designed to work without impacting website performance: think CSP, SRI, HSTS, Referrer Policy, Feature-policy, Trusted Types and Clear-Site-Data.

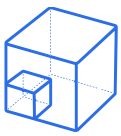
Bottom line: applied correctly, this standards-based approach can actually help *improve* site performance. Who wouldn't want to secure their websites with standards developed by the best minds in the business? Vetted and monitored by organizations like W3C and leading figures in the web security community.

SECURITY, PERFORMANCE AND CONVERSION

Tala's security policies are directly consumed by the browser – in other words, it becomes part of the native implementation itself, meaning there's never any performance degradation to the page loading. Our platform-agnostic approach means you can be up and running in no time.

Zero performance Degradation

Quick integration



No burden on internal infrastructure

Optimized page load times

Tala's innovative solution ensures that all types of client-side attacks are prevented in real time, ***without impacting website performance.*** We do this by automating standards-based security, natively available in every modern browser. This means no overhead and no impact on website performance – in fact, Tala customers report **up to 4% increase in conversions,** with no need for additional instrumentation of installation.

Request your free demo today and see how you can make security work *with* your website performance go