



Webpage Integrity Analysis

Sample Report

Copyright @ 2021

Overview

Webpage Integrity provides you **full visibility of what is happening on the client-side** of your website. This report shows only a sample, based on a single browser session, over a simplified User Flow, to demonstrate some features of the product. In a typical session, Webpage Integrity collects some key metrics:

Domains

97 Total domains receiving data from your web app.

98% Percentage of these domains that are 3rd-party.

Associated risk:

Are you aware of all these destinations, and monitoring for new unexpected destinations?

Assets

468 Total assets being loaded in your web app.

211 Assets that come from 3rd-parties.

80 Total domains loading assets in your web app.

Associated risk:

Are you aware and in control of all these assets? What is the risk of a compromised 3rd party script affecting your website and customers?

Poisoning Events

4 Poisoning events detected on forms.

8 Poisoning events detected on network events.

Associated risk:

Form poisoning allows untrusted code to collect data from your forms. Network poisoning allows untrusted code to intercept network data.

The Client-Side Challenges

The ongoing Digital Transformation is pushing businesses of all sectors to bring to market highly innovative digital products. As development teams are pushed to develop advanced applications in record time, using third-party code has become a standard practice—**66% of modern applications' code comes from third-parties**.

These applications are deployed to adversarial environments, over which the business owner has no visibility or control. This means that the user experience can be compromised, along with the user's information, by exfiltration of sensitive information. This is completely executed on the client-side without the knowledge of the application's backend.

Magecart-like Attacks

“Magecart” is a collective of cybercriminal groups that have been injecting digital **credit card skimmers** on e-commerce and payment websites as early as 2015, but in stronger force since 2018.

The *modus operandi* behind Magecart attacks is to inject malicious JavaScript code into the target company's payment page. This code actively listens to events that happen on the page and collects credit card details whenever a user submits them in a form (event hijacking). These details are then sent to attacker-controlled drop servers, without any awareness from the victimized end-users and companies.

Jscrumbler Webpage Integrity Approach

Webpage Integrity (WPI) is a holistic solution to **detect and prevent malicious code from tampering with the client-side of websites** and leaking sensitive data. To achieve this, WPI detects malicious mutations to the DOM such as the poisoning of event listeners, JavaScript injection, native API poisoning, network requests anomalies, and loaded resources. This detection is achieved by leveraging the browser capabilities and adding verifiers throughout the page.



Report

This report highlights the capabilities of Webpage Integrity in providing visibility over the client-side behaviors on a single session made by us.

During this session, our Embedded Agent (EA) collects detailed information about:

- **DOM Tampering**
- **Code / Native API Poisoning**
- **Network Requests**
- **Inventory with Classification**

On the full report, you can expect high-level tables such as the one below:

Type	Total
Critical - XMLHttpRequest Send	9 
Critical - Form Submit	5 
Other click events	523
Critical - button click	74
Other keyboard/mouse events	8

Plus, it will contain several other tables and graphs with detailed information on specific detections such as **code/native API poisoning**, **source domains**, **destination domains**, **resources/scripts**, **breakdown of scripts** into first and third party, and a **digested overview of the critical issues**.

Inventory

With our Inventory feature, you get an overview of **every resource and network asset** that has been loaded or performs changes on the web page. The table below displays how many third-party assets are being loaded, broken down by domain:

Resource	Count
s.w.org	2
res.cloudinary.com	3
cdn.segment.com	1
cc.swifttype.com	1
ads.avct.cloud	1
s7.addthis.com	2
cdn.cookieclaw.com	6
cdn.ampproject.org	4
static.intercomassets.com	1
js.intercomcdn.com	2
intercom.io	3
gstatic.com	2
googleoptimize.com	1
doubleclick.net	2
www.google.com	1
www.googletagmanager.com	1

The table below displays a sample shortlist of the inventory:

Resource	Third-Party	Domain
s.w.org/images/example.svg	YES	s.w.org
res.cloudinary.com/js/example.js	YES	res.cloudinary.com
cdn.segment.com/js/example.js	YES	cdn.segment.com
cc.swifttype.com/js/example.js	YES	cc.swifttype.com
ads.avct.cloud/js/example.js	YES	ads.avct.cloud
s7.addthis.com/js/example.js	YES	s7.addthis.com
cdn.cookieclaw.com/js/example.js	YES	cdn.cookieclaw.com
cdn.ampproject.org/js/example.js	YES	cdn.ampproject.org
static.intercomassets.com/js/example.js	YES	static.intercomassets.com
js.intercomcdn.com/js/example.js	YES	js.intercomcdn.com
intercom.io/js/example.js	YES	intercom.io
gstatic.com/js/example.js	YES	gstatic.com
googleoptimize.com/js/example.js	YES	googleoptimize.com
doubleclick.net/js/example.js	YES	doubleclick.net
www.google.com/js/example.js	YES	www.google.com
www.googletagmanager.com/js/example.js	YES	www.googletagmanager.com
your-website.com/content/js/script.js	NO	your-website.com
static.your-website.com/shared/example1.js	NO	static.your-website.com
assets.your-website.com/shared/example2.js	NO	assets.your-website.com
assets.your-website.com/shared/example3.js	NO	assets.your-website.com

You can clearly see multiple assets loaded from third-party domains that were present in the session.

We also map out all Resource Assets that correspond to the Domains or URIs responsible for generating events that triggered a detection by WPI's monitoring, such as **DOM mutations** (e.g. adding a script to a page or modifying an attribute in a form) and **poisoning** of native browser functions (e.g. modifying the function of an onSubmit event).

Contact Us

If you want further details about how Jscrambler can help you address a specific use case, don't hesitate to contact us

hello@jscrambler.com

+1 650 999 0010

The Gartner logo, featuring the word "Gartner" in a bold, blue sans-serif font with a registered trademark symbol (®) to the right.

Jscrambler is the leader in Client-Side Application Security

Recognized in **Gartner's Market Guide for Online Fraud Detection**

and in **Gartner's Market Guide for In-App Protection**