Global Data At Risk

State of the Web Report

Summary

2020 has been an extraordinary year. This report is offered during a time of unprecedented change that is transforming the way we work, shop, do business or access healthcare and financial services. The COVID-19 pandemic has changed everything - and provided a more extensive threat vector for cybercriminals to exploit. This report, conducted by analyzing the architecture and integrations of the 1000 most frequently requested websites from Alexa 1000, is intended to shed light on how vulnerable websites and web applications are today.

The key findings of this year's Global Data at Risk - 2020 State of the Web Report suggest there's a lot of room for improvement. As expected, the data illustrates an increased reliance on JavaScript integrations to power websites. Surprisingly, security efficacy has declined.

Just 1.1% of websites analyzed were found to have effective security in place - an 11% decline from 2019. While deployment volume went up, effectiveness declined more steeply. Attackers have the upper hand mainly because we are not playing effective defense.

Many of you may read this report because you've realized you have Magecart risk. You do. However, this report is intended to shed light on the fact that the problem is significantly bigger than Magecart. After all, what's at stake is significant, as continued attacks risk the erosion of the most important ingredient that powers e-commerce and digital transformation: trust.

Data, the most significant driver of the digital economy, is at risk.

Key findings from the Global Data at Risk - 2020 State of the Web Report highlight the scope of vulnerability and that the majority of global brands fail to deploy adequate security controls to guard against client-side attacks:

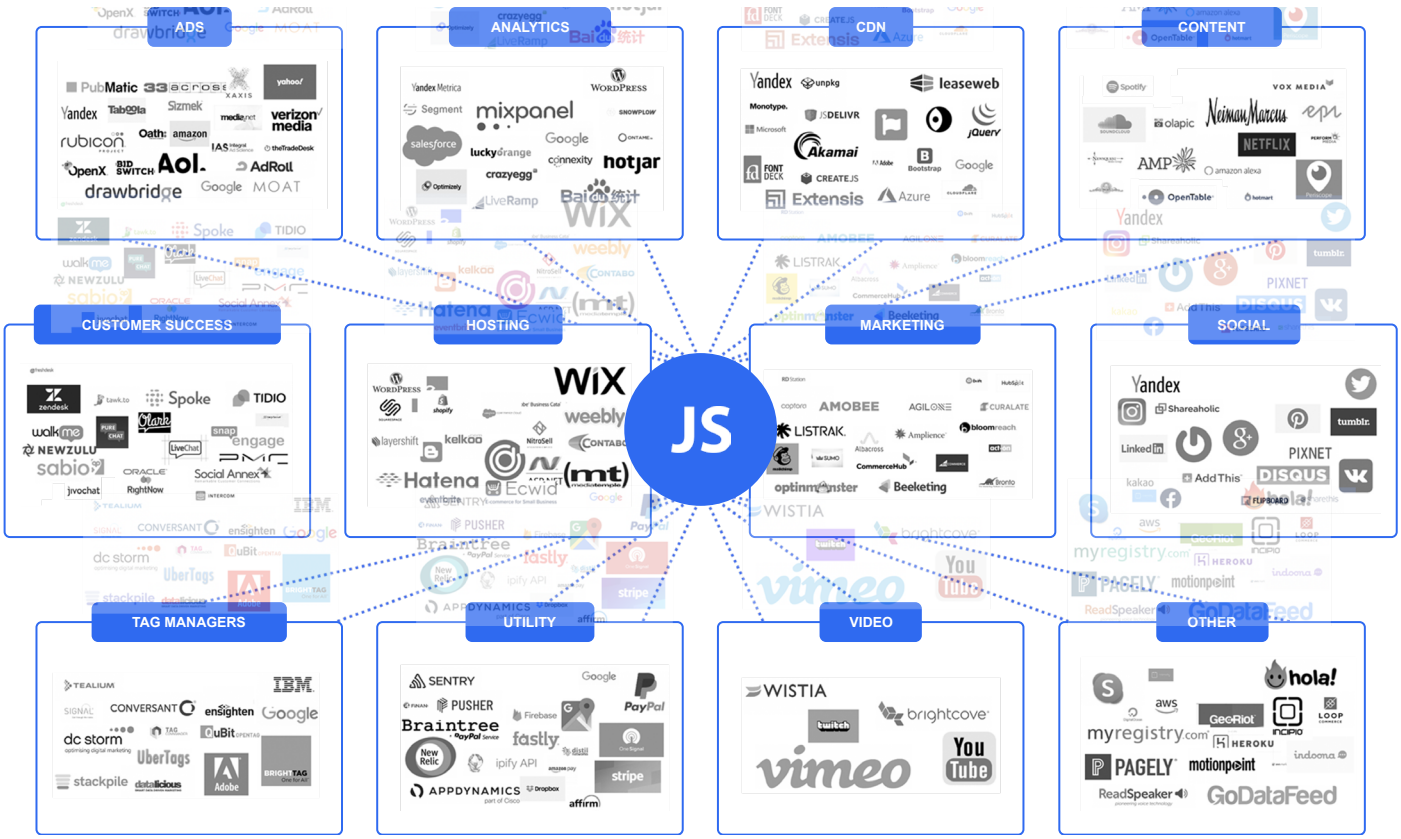
- Despite increasing numbers of high-profile breaches, forms, found on 92% of websites expose data to an average of 17 domains. This is PII, credentials, card transactions, and medical records. Tala's analysis shows that this data is exposed to nearly 10X more domains than intended. Nearly one third of websites studied expose data to more than 20 domains.

- COVID-19 has increased reliance on digital experience. 58% of the content that displays on customer browsers is delivered by third-party JavaScript integrations. This website supply chain leverages client-side connections that operate outside the span of effective control in 98% of sampled websites. The client side is a primary attack vector for website attacks today.

- Over 99% of websites are at risk from trusted, whitelisted domains like Google Analytics. These can be leveraged to exfiltrate data, underscoring the need for continuous PII leakage monitoring and prevention, similar to DLP for web applications. This has significant implications for PII leakage and, by extension, GDPR and CCPA.

- No attack is more widespread than Cross-site Scripting (XSS) but 97% of websites are using dangerous JavaScript functions that could serve as injection points to initiate a DOM XSS attack.

- The average website in the Alexa 1000 creates 10 connections to the user's browser - a 21% increase on 2019.



99% of websites globally include multiple client-side vulnerabilities, making them attractive targets for attackers

Introduction

The good, the bad, the unpredictable

The global pandemic has driven an unprecedented increase in online transactions and e-commerce. By mid-April alone, US retailers' online YoY revenue growth was up 68%, with a 146% growth in all online retail orders[1]. As millions of people worldwide are affected by stay-at-home orders, we're beginning to see a dramatic shift in consumer behavior towards everything-online. While many have observed that the digital disruption in commerce is inevitable, this crisis has accelerated that trend.

Enterprise websites have long been a major target for attackers in search of credit card data, PII, credentials, healthcare information, malicious ad injection, session hijacks and re-directs and more. With e-commerce sales projected to grow, client-side web vulnerabilities pose a unique, global threat to consumer data security - and the businesses associated with any breach:

- A data breach can wipe as much as 7.2% off a company's share price: around £8.8m for UK-listed companies or up to $32.3m for US companies.[2]
- The average cost of a breach in 2019 was $3m per organization, with lost business accounting for 36% of the cost - consistently the largest cost factor, driven by an average customer turnover rate of 3.9%.[3]

From a data privacy and protection perspective, increasing use of vulnerable forms or unmanaged third/fourth/fifth+ parties can bring additional challenges for businesses seeking to comply with regulations like GDPR or CCPA. Many third-party integrations collect and re-purpose customer data collected from your site - anything from geolocation information to cookies and storage. And, as a recent manipulation of Google Analytics has shown, without constant monitoring, even legitimate, whitelisted applications are vulnerable to data exfiltration and leakage. Increasingly, data privacy mitigation will involve answering the question: "Do you have a handle on the data your third-party integrations are collecting?"

For e-commerce, the switch in purchase drivers presents a real opportunity for excellence in online experience. With more online shoppers than ever, site performance, frictionless transactions and security can become key differentiators, not least because attackers view this crisis as a great opportunity:

- **Magecart attacks on online retailers and banks increased by 20% during the pandemic.[4]**
- **Web applications were involved in 43% of breaches in 2019 ; 37% used or stole credentials.[5]**

While the pandemic has seen cybercriminals adapt to take advantage of changing circumstances, this analysis indicates that owners of the world's top websites have been slower to respond to the accelerating threat landscape.

Last year, Tala Security's analysis of the Alexa 1000 websites revealed that only 2% of the world's leading websites were using security controls and standards capable of mitigating client-side risks.

One year later, very little has changed; if anything, it seems security is headed in the wrong direction. This is a particularly troubling finding given that website owners will be increasingly driven to improve user experience and enhance website richness in response to the dramatic shift towards online commerce.

[1]CC Insight: US retailers see online growth YoY in April similar to recent holiday season.
[2]Data Economy: Data Breaches Wipe 7.2% off Average Company Share Price
[3]IBM: Cost of a Data Breach Report 2019.
[4]Wired: Online Credit Card Skimmers are Thriving During the Pandemic
[5]Verizon: 2020 Data Breach Investigations Report

2019

June
- Quest Diagnostics & LabCorp

May
- Mirrorthief
- Picreel & Alpaca Forms
- Cleor
- Forbes
- Leicester City FC

April
- AeroGrow
- Atlanta Hawks
- Puma Australia

July
- The Guardian
- Pelican

August
- Everlast
- National Baseball Hall of Fame
- Poker Tracker

September
- Garmin SA
- SEPTA
- Fragrance Direct

December
- Smith & Wesson
- Sweaty Betty
- Rooster Teeth Productions
- Missoma

November
- Macy's

October
- Sesame Street
- Mission Health
- Umbro Brasil
- First Aid Beauty
- American Cancer Society

2020

January
- WhatsApp vulnerability

February
- EuroTickets2020
- OlympicTickets2020
- Hanna Andersson
- Focus Camera
- Blue Bear
- Australian Bushfires
- Donation Website

March
- NutriBullet
- Tupperware
- Truefire

June
- Fitness Depot
- Claire's
- Intersport

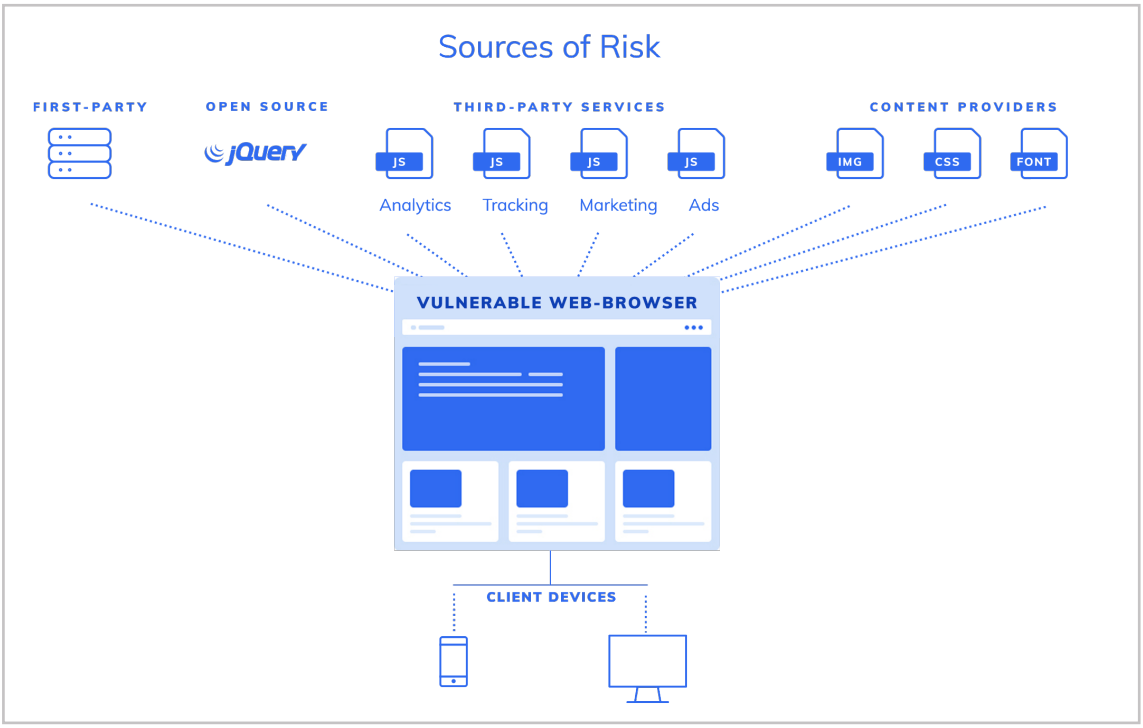
May
- Paramo
- EasyJet

April
- Robert Dyas

Website Client-side security risks

One vector, every sector

Modern web applications and websites have undergone key architectural shifts, with important security implications. Today's websites integrate code and resources from dozens of third-party service providers, including user analytics, marketing tags, CDNs, and third-party JavaScript libraries. These integrations provide a rich, dynamic web experience, compelling content and enhanced capabilities. Unfortunately, they also introduce critical vulnerabilities that enable client-side website attacks.
In the absence of correct vetting, this third-party executable content can potentially be compromised or malicious. These attacks typically involve credential theft, PII or credit card details, stolen via a 'skimming'-type attack - widely known as 'Magecart' attacks. It seldom happens in a single 'mugging' incident involving an individual user's data; by compromising a widely used third-party component, attacks like these can 'mug' multiple users on as many different sites as they wish, all exploiting the same vulnerability. But that's not even the whole story: there are many different ways hackers can exploit JavaScript to access sensitive data - and many types of data they can steal, from personal data to geolocation information. From a data privacy and compliance perspective, even legitimate third-party integrations can have unintended negative consequences if they gather and share your customers' information without your knowledge.
It doesn't matter what sector you operate in, JavaScript vulnerability impacts all websites globally. It's incumbent on website owners to secure the user experience on the browser sessions presented to end users.



This 'under the hood' view of a typical enterprise website's composition illustrates the large number of integrations, servers and domains that inject code, content and enable data collection from your customer's browser sessions.

The weak links in the website supply chain

Attackers like Magecart have increasingly targeted this architecture, known as the 'website supply chain' because it presents an attractive, vulnerable entry point for hijacking customer browser sessions and accessing their data. What makes it even more lucrative is scalability: one attack has potential for massive reach, enabling attackers to access thousands of sites simultaneously.

The threat posed by complex and often opaque supply chains is growing; all too often, organizations don't have a handle on just how many of these integrations are running on their websites, never mind who/how their security is being managed. What that means in real terms is that customer data is at risk every time they log in or enter their personal information, fill out a form or simply visit your website.

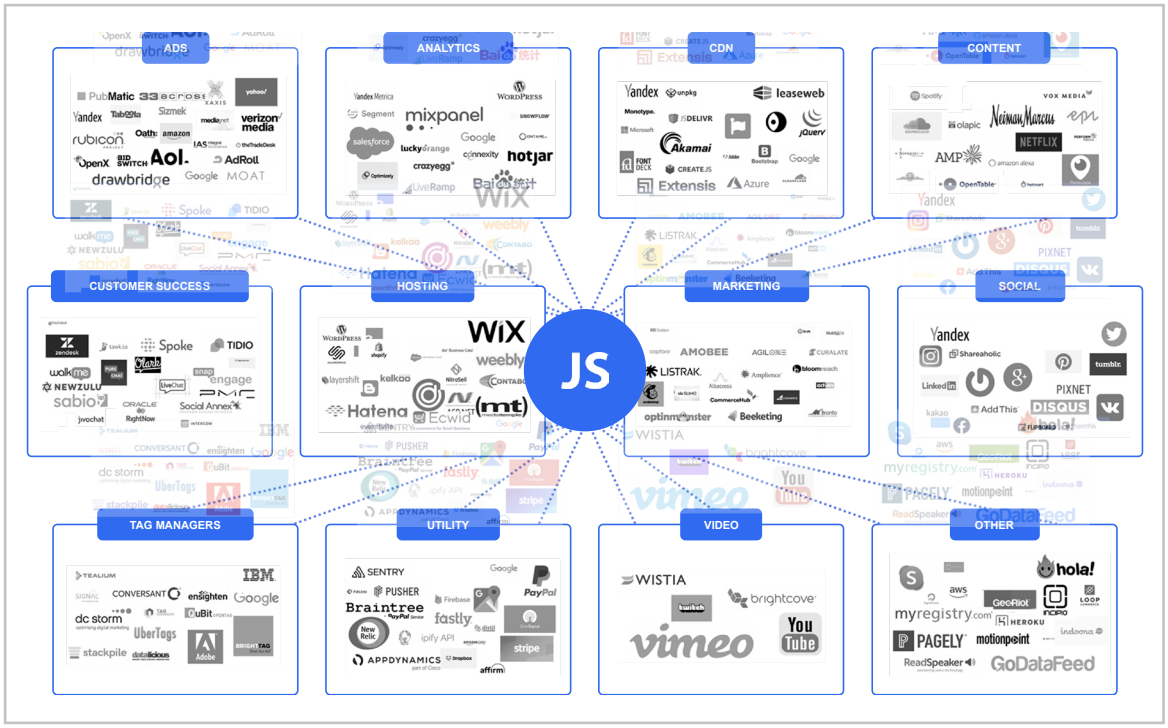
Attacks targeting the vulnerabilities inherent in every modern web architecture have many names, including Cross-site scripting (XSS), Formjacking, Magecart, Ad injection attacks, Content injection Attacks, Cryptojacking and Website Supply Chain attacks.

The vulnerabilities they exploit include:

- First party services
- Third party vendors - the 'website supply chain' Open source JavaScript libraries
- Ad servers - this attack amplifies the problem exponentially into 4th, 5th parties
- Network entry points serving content, such as Amazon S3 buckets.

The scope of potential damages resulting from these attacks include:

- Data Skimming
- Payment Card Skimming
- Formjacking
- Keylogging
- Screen Scraping
- Clickjacking
- Phishing
- Web Injection
- Ad Injection
- Content Injection
- Session Redirect
- Form Field Manipulation
- Defacement
- Malware, Banking Trojan, Ransomware Distribution, Cookie Sniffing



99% of websites globally include multiple client-side vulnerabilities, making them attractive targets for attackers

Risk Analysis: Javascript

The language of the dynamic, rich web - and cybercriminals

The modern web is powered by JavaScript (JS): it operates through client-side connections on 96% of the world's websites, driving the dynamic, rich experience and analytics behind every customer-focused website. As much as 70% of the code rendering on websites today comes not from the site owner's server, but via JS integrations operating outside the security controls the majority of site owners deploy.

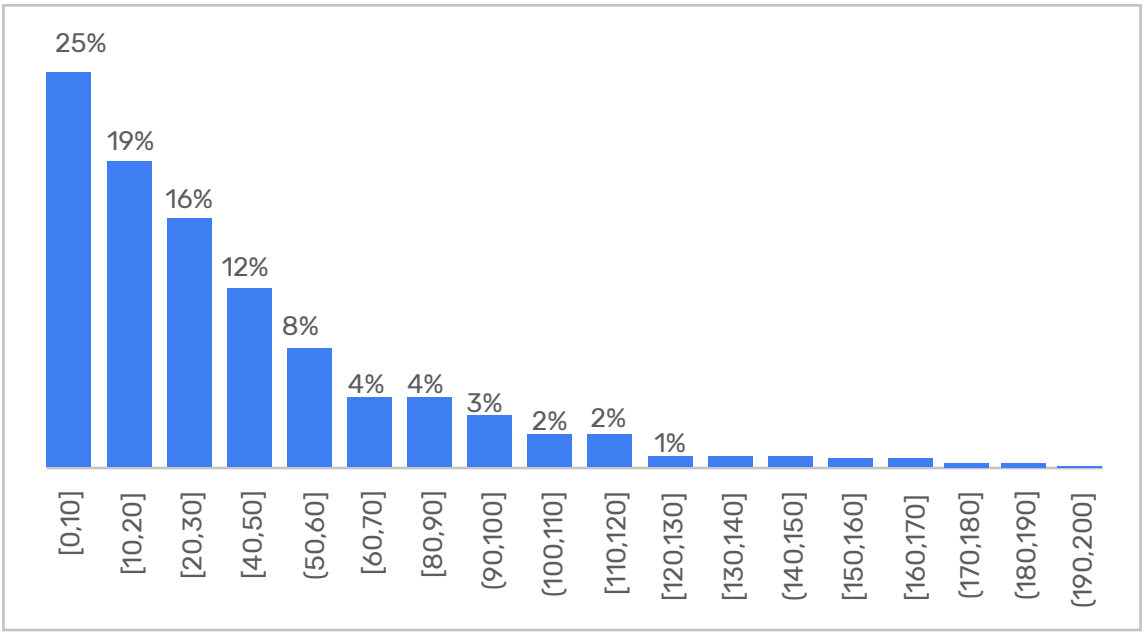
Unfortunately, because these integrations are largely unmanaged and unmonitored, they substantially expand the attack surface, introducing significant risk to both the business and its end users. These integrations have done a great job of increasing functionality - but without bridging the security gap they have opened up. And that's the gap that cyberattackers like Magecart are specifically looking to exploit. In some of the most prominent website attacks and data breaches of the past two years, typically labelled as 'Magecart', malicious JS code is inserted via a compromised third party, facilitating PII theft, payment card skimming or other data theft.

Third-party risks have increased in 2020

In 2019, Tala's analysis of the Alexa 1000 websites showed a growing and large dependency on third-party services. In 2020, that number has increased to 32 third parties operating on the average Alexa 1000 website.

Each 3rd party represents a unique domain outside the security scope of the Alexa 1000 company hosting it, i.e. a potentially uncontrolled risk to data security and attack.

As the histogram below illustrates, over a third of website owners integrate more than 30 third parties. Although this volume of integration is intended to provide a rich website experience, without a corresponding increase in client-side security, the risk introduced by this website supply chain fair outweighs any benefit they may provide. For many website owners, the endeavour backfires: breaches like magecart result in significant costs, including mitigation, fines, brand damage,



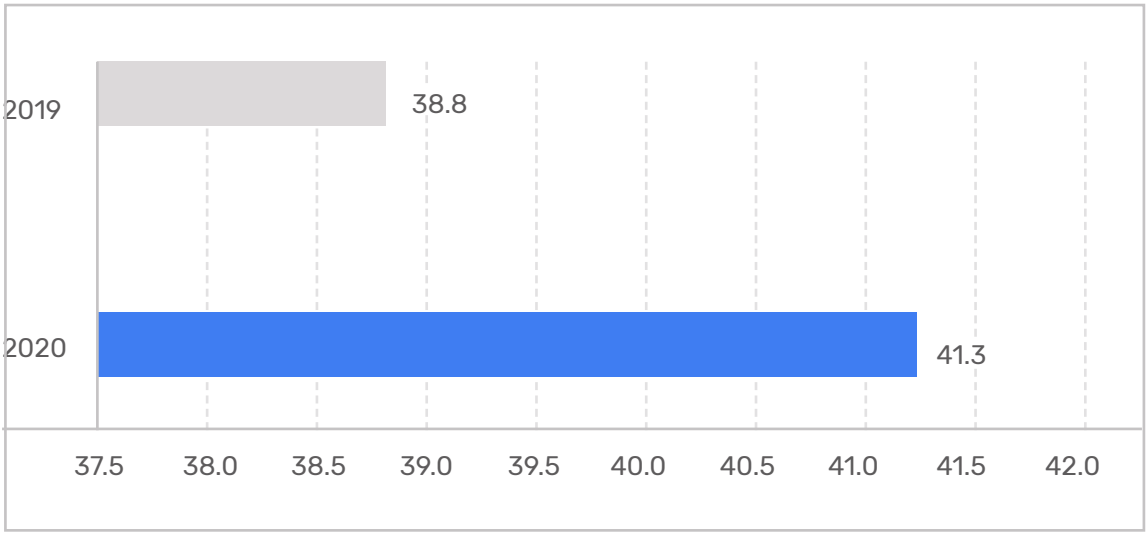
The average Alexa 1000 website relies on 32 third-party integrations.

class-action lawsuits and customer churn. Despite high-profile attacks on global brands including NutriBullet, Tupperware and Páramo, the need to secure these integrations effectively is not cutting through to enterprise website owners. In an increasingly stringent regulatory data protection environment, the onus is on website owners to protect users from these breaches.

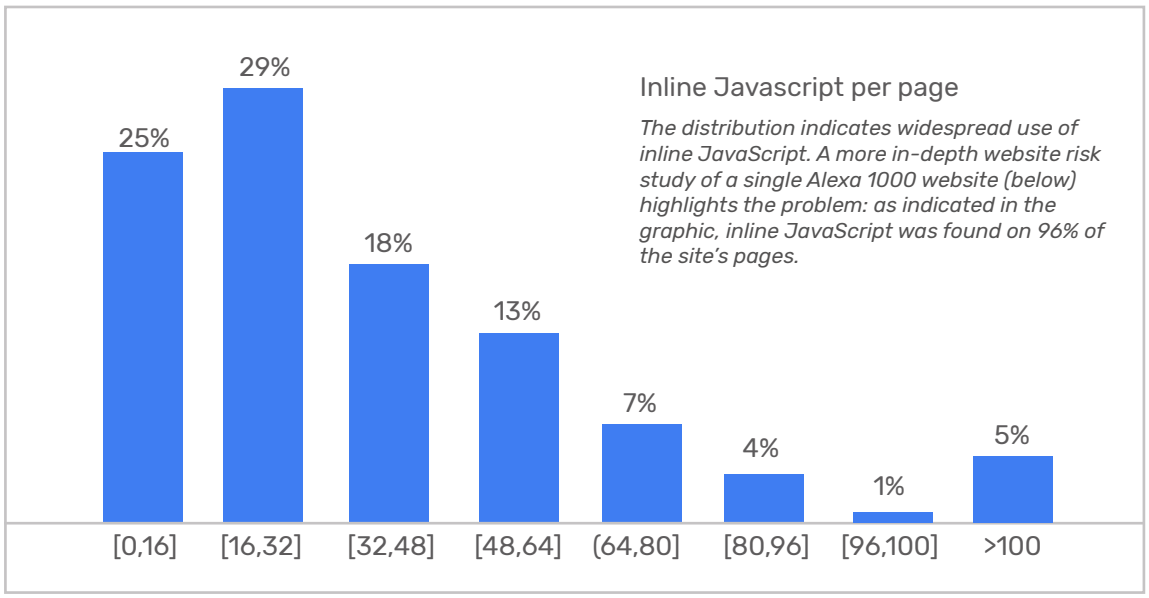
Inline JavaScript

JavaScript can be executed as inline scripts or externally loaded. Because inline scripts are more difficult to read - and therefore more difficult to maintain - application security teams are increasingly asking developers to eliminate the use of inline scripts in favor of loading JS via external files (see following sections on externally loaded scripts). When code is harder to read, it's easier for hackers to hide malicious code. Because inline JavaScript is loaded from trusted domains, protection against malicious attacks requires the use of advanced security controls, such as CSP-based nonces.

So how are the AppSec teams for the Alexa 1000 websites doing in their fight against inline JavaScript? Not great: 2020 has seen a 6% increase in its use. An average of over 41 inline JavaScripts were found.



Despite the risks, Alexa 1000 websites have increased the use of inline scripts by 6% since 2019.

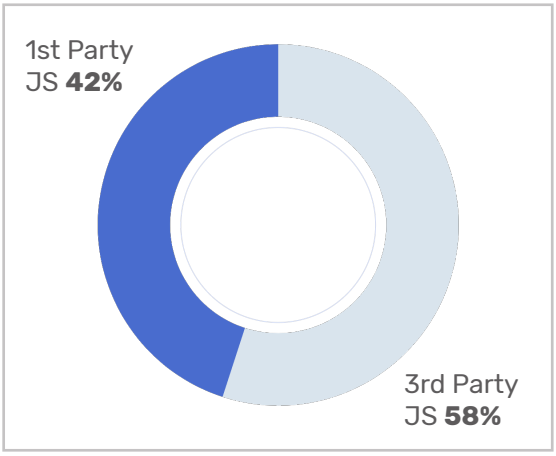


We're still not learning: External JavaScript - First vs Third-party loaded scripts

Externally loaded JavaScript files(3rd party JavaScript) are one of the primary attack vectors exploited by hacker groups like Magecart. To mitigate the risk, it's vital to monitor and control their use. It's also important to note that attacks can be launched from compromised first party domains as well as third parties - as major global brands like British Airways, NutriBullet and Ticketmaster have found out at significant cost[6].

Our analysis for 2020 reveals that, despite multiple high-profile attacks, Alexa 1000 sites have actually increased the number of first and third-party scripts they load:

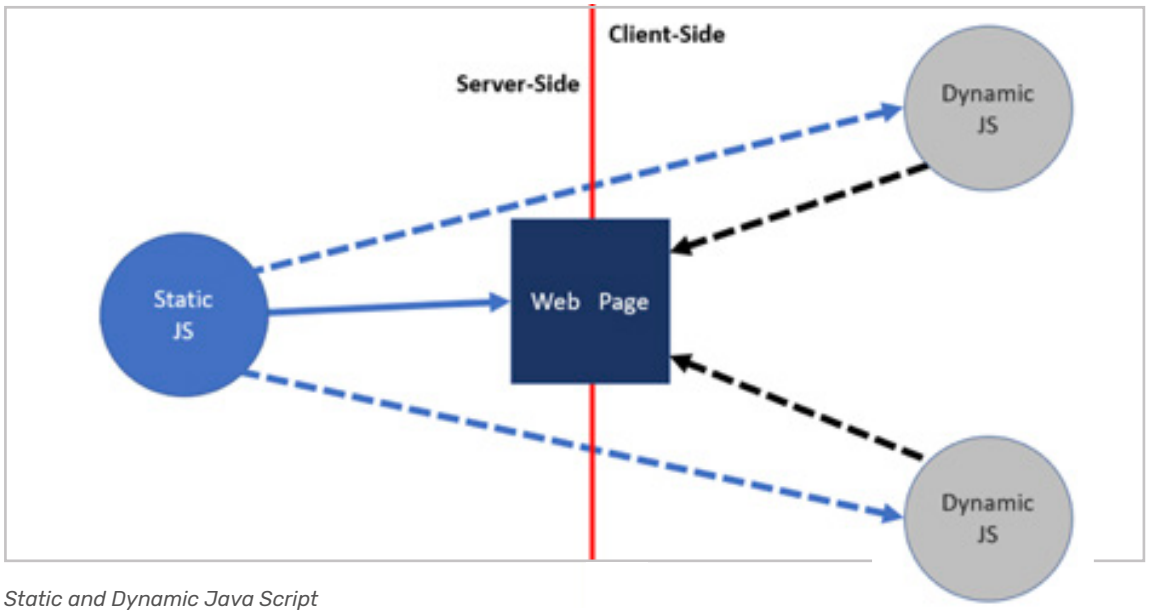
- 35.4 first party scripts, a 30% increase over 2019
- 32 third party scripts, a 14% increase over 2019.
- Overall, the study found a 20.5% increase in external scripts



2020 External Javascript: 1st Party vs 3rd Party

External JavaScript: Static vs Dynamic

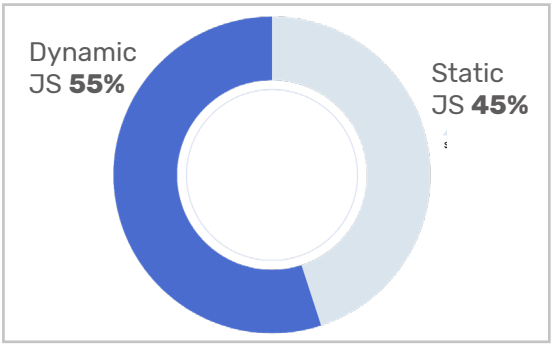
Modern websites are dynamic. That's great for the user journey, but not so good for website security. On almost all sites, the external JavaScript code that drives these enhancements isn't loaded directly from the web server. In this case, static JavaScript code delivered from the web server dynamically calls additional code that's delivered to the browser via uncontrolled client-side connections. The modern web relies heavily on this architecture. In the past, the bulk of the logic behind web applications was hosted on the server. Today's more powerful web experience is powered by a shift towards rich client-side applications that use dynamic calls to JavaScript for a significant portion of their functionality within the browser.
This 'piggybacking' creates a significantly more expansive surface for hackers to exploit. As more scripts are added and



Static and Dynamic Java Script

[6]British Airways breach: how did hackers get in?

connect to others, you can end up with a daisychain effect. The TicketMaster Magecart breach in 2018 is attributed to malicious JavaScript piggybacking on the site's integrated chatbot service. Because both static and dynamic JavaScript share the same privilege level, the attack surface rapidly expands, providing multiple paths for malicious actors to access your data, and significantly amplifying the risk posed by code the website owner doesn't control. As the chart below shows, more than half the JavaScript rendering on the customer's browser when they visit Alexa 1000 websites is loaded via dynamic JavaScript.



2020 Static vs Dynamic Javascript

Cross-site Scripting (XSS)

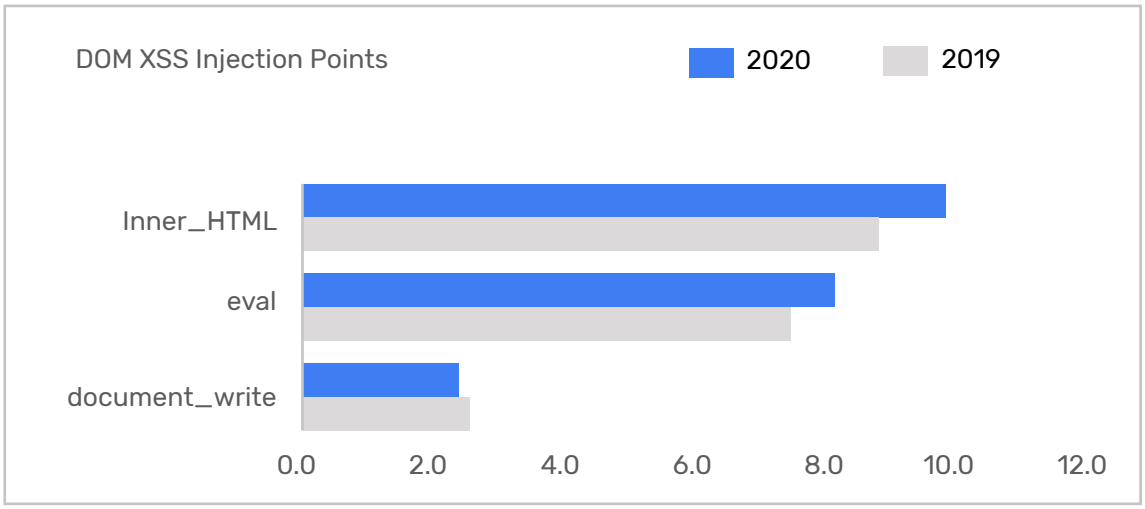
Cross-site Scripting (XSS) is the most prevalent type of client-side attack. Although other client-side attacks, such as Magecart, formjacking and card skimming, feature more prominently in the news, no attack is more widespread than XSS.

In a DOM XSS attack, a hacker is able to execute malicious code inside the browser. Because the malicious code is launched directly in the browser, these attacks are particularly difficult to detect and block. DOM-based XSS vulnerabilities usually arise when JavaScript takes data from an attacker-controllable source and passes it to a sink that supports dynamic code execution, such as eval() or innerHTML. This allows attackers to execute malicious JavaScript to modify the Document Object Model (DOM).

The DOM is a map for how a webpage renders and functions. The ability to control and/or modify the DOM is attractive to hackers because it provides complete control over the website experience and any data entered or rendered there.

97% of websites use dangerous JavaScript functions

This study has identified an increasing number of DOM XSS risks. The graph below shows the incidence of potential injection points, such as document.write(), eval(), and innerHTML() among the study sample. A significant majority of websites are using dangerous JavaScript functions that could serve as injection points to initiate a DOM XSS attack; the innerHTML function was found operating on 97% of websites.



DOM XSS Injection Points

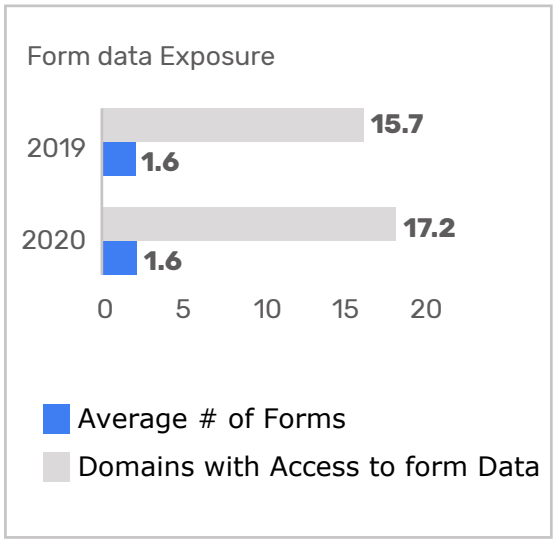
Risk Analysis: Form Data Exposure

Unintentional over-exposure, significant risk

92% of the Alexa 1000 use forms to collect user information, including login forms, search and user input fields. Groups like Magecart target these forms to steal user-entered data, including credentials, financial information, PII, healthcare records and other sensitive information.

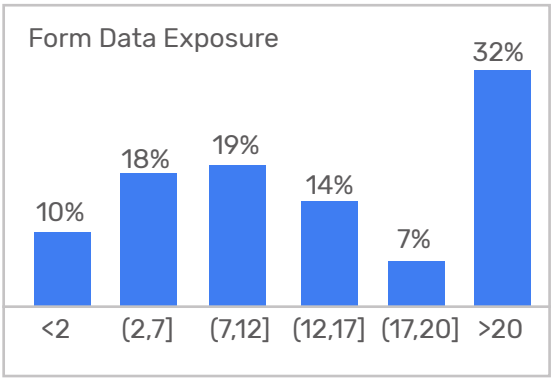
To understand the risk, the graph below shows a histogram of the Alexa 1000's "Exposure Ratio": the number of domains with access to form data on a website relative to the number that should have access to it. The higher the number, the greater the exposure.

As the below graph illustrates, Tala's analysis reveals a significant increase in form exposure risk for Alexa 1000 websites in 2020:



Form Data Exposure

This shows shows that, while form data is defined by the website's architecture to be shared with an average of 1.6 domains, in reality, that data is exposed to an average of



Cross Site Scripting

User form data is exposed to an order of magnitude more domains than intended by Alexa 1000 website owners.

17.2 third-party domains: an increase of 9% on 2019's numbers.

The histogram above paints an even more alarming picture, as almost a third of Alexa 1000 websites expose form data to more than 20 domains. Clearly this is an unintended, but nonetheless uncontrolled, problem. This order-of-magnitude increase in exposure translates into a massive security risk for all website owners: these uncontrolled, unmanaged access points are fertile hunting ground for Magecart and other attackers and will and have resulted in significant data privacy fines.

Risk Analysis: Third Party Content Integration

Integrating vulnerability into websites

JavaScript is the workhorse for the web. The average Alexa 1000 website relies on 32 third parties to render browser sessions for their customers, suppliers and employees, including images, stylesheets, fonts, media, iFrames and more.

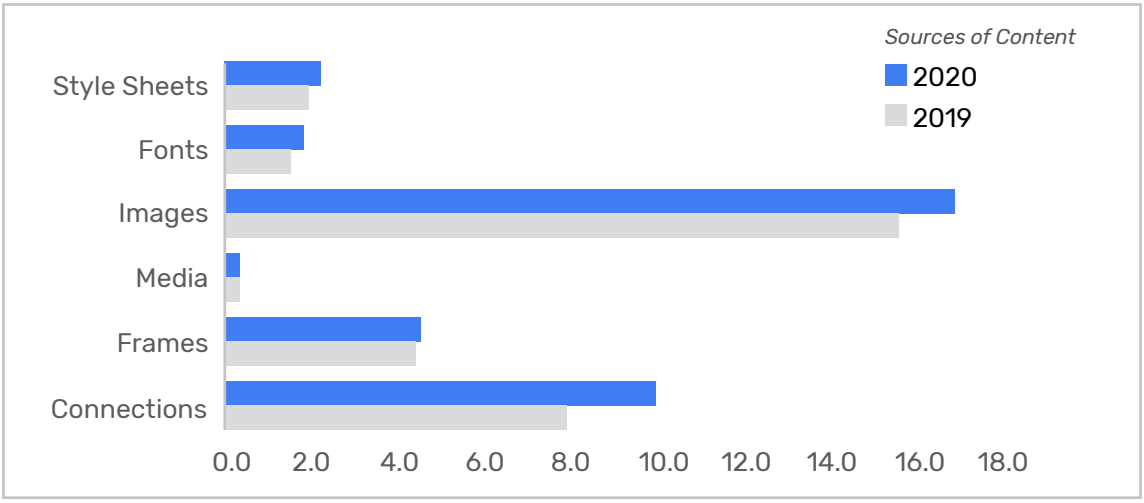
Unsurprisingly, content sources have increased along with website richness. Some of this content is overlooked, providing opportunity for attackers.

Images, stylesheets and fonts can carry hidden dangers

Hackers often try to serve malware to users via unwanted or malicious images, fonts and stylesheets - all important aspects of website and web application functionality, but often overlooked from a security perspective. For example, cascading style-sheets (CSS) injection attacks can be used to exfiltrate user data such as passwords, credit card details and other PII. 'CSS Exfil' can also be used to de-anonymize users on dark nets such as Tor. Magecart attacks often exploit XHR requests required by these third-party integrations to launch attacks and exfiltrate users data from the browser.

As CSS and other browser-native tools develop and evolve to become even more functional, they'll become even more attractive to attackers looking to bypass security controls.

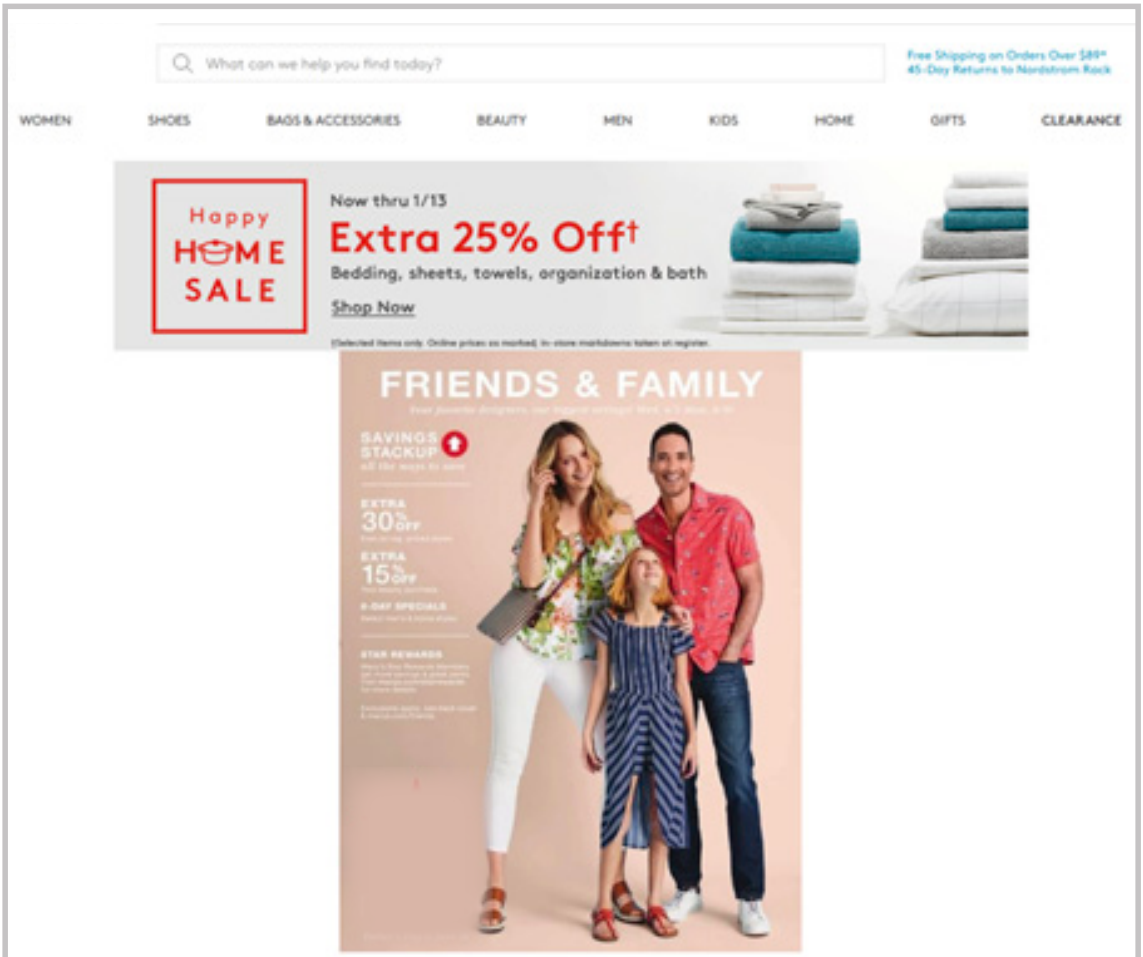
The average website in the Alexa 1000 creates 10 connections to the user's browser - a 21% increase on 2019. These include XML requests, websocket connections and eventsource interfaces. All are potential points of vulnerability to be exploited.



iFrames: Stealing your customers, degrading user experience

As the 'sources of content' graph above shows, iFrame injection is a common attack vector. Hackers insert malicious iFrames onto websites to display competitor or other unauthorized ads, insert malware links or divert users to malicious sites serving attacks or unauthorized content.

These attacks aren't focused on stealing customer PII - they're focused on stealing your customer and using them to bolster traffic. These attacks cause lower conversion rates, higher cart abandonment and a degraded user experience. They also impact in other, unexpected ways: an estimated 80% of display ads are bought programmatically[7], meaning that the threat posed by complex and often opaque supply chains is growing. Advertisers displaying a million ads over a 24-hour period are likely to pay for more than 100,000 ads before any issue is detected.[8]



Malicious or competitive, non-approved ads can be served to end-users. As this mocked-up screenshot shows, the website the end user is browsing to (retailer.com) is serving ads for a competitor (retailer-competitor.com)

[7] R. Benes, "Agency Pros Say Fraud is Biggest Threat to Their Budgets", eMarketer.com 22 February 2019. https://www.emarketer.com/content/agency-pros-say-fraud-is-biggest-threat-to-their-budgets

[8] Traffic Guard & Juniper Research "Digital and mobile ad fraud to cause losses of $100 million by 2023 for North American advertisers" 18 February, 2020 https://www.businessofapps.com/news/digital-and-mobile-ad-fraud-to-cause-losses-of-100-million-by-2023-for-north-american-advertisers/

Risk Analysis: PII & Data Leakage

It's all about the data

Data privacy and protection are fundamental aspects of every business with a digital footprint. From GDPR to CCPA, HIPAA, PCI-DSS, APP (Australia) and PIPEDA (Canada), compliance isn't just about meeting legal obligations; as many businesses have learned, breaches cost money in fines, reputational damage, compensation claims and lost business.

Most businesses are aware of their obligations and have implemented strategies and cybersecurity solutions to help secure their data. But some key assets are often overlooked, as Tala's research has found, including websites and web applications with an average of 49 integrations.

It's not just about customer-entered data either; website owners are directly responsible for the protection of all PII on their websites, so it's worth taking a look at some of the often-overlooked risks:

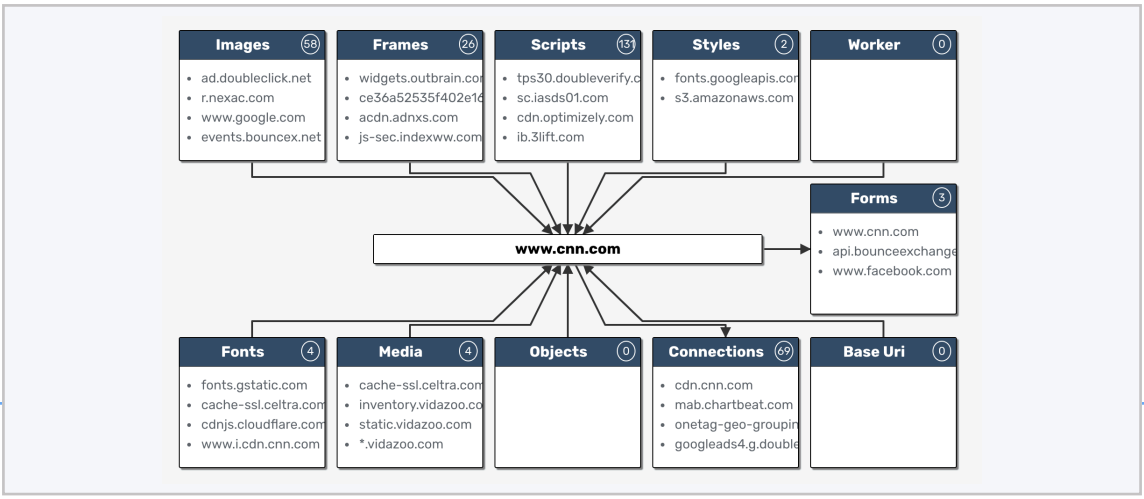
- **Cookies:** Sensitive data stored in persistent cookies (such as username and password) is vulnerable to exploit and can be used to enable attacks such as session hijacking, account takeovers and PII breaches.

- **Web storage objects can be exploited:** capable of storing up to 5MB of information that can survive a page refresh or even a browser re-start, locally stored web objects can be accessed and compromised via any JavaScript running on your page.

- **JSONP endpoints:** When present on the main website or integrated third-parties, these can potentially allow an attacker to circumvent a CSP policy, because JSONP endpoints bypass the Same-Origin-Policy of the browser.

- **Trusted/whitelisted domains:** Even trusted domains, such as Google Analytics, are vulnerable to compromise causing data leakage or theft. Over 99% of websites are at risk from this: whitelisting enables data access, but verification and authorization mechanisms don't exist, meaning PII theft and data exfiltration can occur without the owner's knowledge – with obvious implications for GDPR and CCPA. PII Exposure Scanning and PII Leakage Mapping provide the proactive monitoring and granular control needed to mitigate this risk. To be effective, monitoring and data leak prevention must happen in parallel: simply knowing that third-party services are capable of collecting data is not the same as preventing unwanted exfiltration. The ability to control data accessibility to both trusted and untrusted domains is critical – and largely missing from the security controls implemented by the Alexa 1000.

Analysis of xhr requests will discover the JavaScript libraries that have access to sensitive data, giving the flexibility to control access. To develop a complete view into sensitive data exposure, including stored in local or session storage, it is critical to track and restrict all of these components.



Closing the client-side security gap with website security standards and controls

You can't secure what you don't know is there

Given the significant architectural shift in website and web application infrastructure, server-centric or network-centric application security such as Web Application Firewalls (network WAF) and Runtime Application Self-Protection (network RASP) have little visibility into third-party integrations on websites and web applications.

In many instances, enterprises don't even have visibility into what third parties are integrated into their website and mission-critical applications. They have limited/no ability to identify and block compromised third parties. Traditional application security models don't integrate well with the "continuous delivery" model adopted by most enterprises, where applications are updated on a weekly, sometimes daily, basis.

The same web development experts that created today's rich, powerful browser experience understood that all the extra functionality created a gap in security. No cohort of technologists is better placed to see - and mitigate - browser-based risk. From Google to the W3C, powerful, fine-grained security controls are constantly being developed and embedded in modern browsers.

These browser-native security controls include:

- Content Security Policy (CSP)
- Subresource Integrity (SRI)
- Strict Transport Security (HSTS)
- Referrer-policy
- Feature-policy
- Trusted Types
- iFrame sandboxing rules
- Certificate Stapling
- Clear-site data

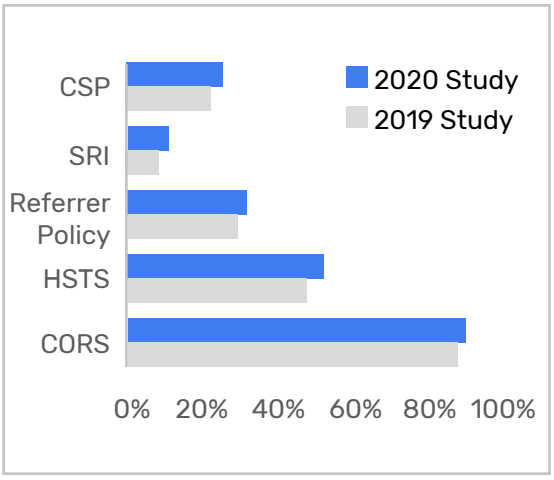
These standards and controls protect against a wide range of attacks, including cross-site scripting, clickjacking, iFrame injection and Magecart-style supply-chain attacks.

Implemented together, they provide powerful, comprehensive, future-proof protection for today's web and are recommended by leading organizations like the PCI Council and RH-ISAC (both of which recommend CSP).

Which is why every enterprise deploys them, right? Not so fast…

The State of the Web 2020: From a security perspective, more miss than hit

Let's start with the good news: security policy deployment is on the increase. CSP alone has increased by 11%, while more than half of the Alexa 1000 websites have deployed HSTS. CORS (cross-origin resource sharing), is still the only widely adopted standard.

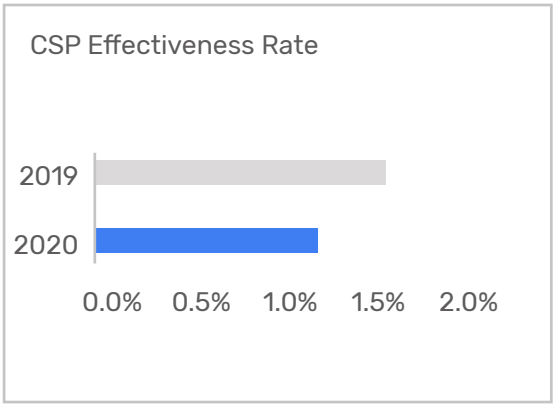


Security policy deployments

Content Security Policy (CSP)

Just over 30% of sites have deployed some form of Content Security Policy. Unfortunately, the effectiveness of those implementations is low: 35% less than in 2019, coming off an already-low bar. Just 1.1% of Alexa 1000 sites that have implemented it have effective CSP in place. To put it another way: almost all sites that have implemented CSP have policies in place that offer little or no protection. Despite the fact that these controls are

Only 1% of Alexa 1000 websites have deployed effective protection against Magecart-style attacks.



literally built-into the browser, website owners continue to struggle to implement them effectively. This is because they can be complex and time-consuming for security teams - a challenge compounded by the dynamic nature of today's websites. Get it wrong, and you break the site without really doing anything to improve security. To make things more difficult, CSP's capacity to deliver valuable insight into website attacks and behaviors can become alert-overload and fatigue for security teams that aren't equipped to deal with them. In these situations, automation offers a simple, highly effective solution.

Subresource Integrity (SRI)

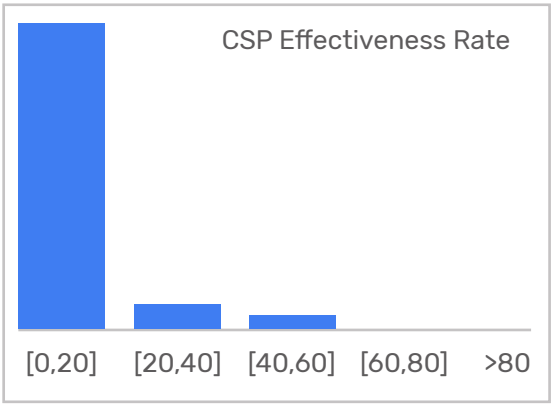
As we've seen, the majority of externally loaded JavaScript on the Alexa 1000 comes from third-party integrations, 58% in 2020. Subresource Integrity delivers highly effective protection against malicious externally loaded JavaScript, including integrity hashes (akin to code signing) - but only 6% of websites use it.

Referrer-policy

Any time a user clicks on a link on a site and that link is to an external domain, the external domain receives information about the originating one. This is really useful for helping us track how users landed on our sites but can become a source of data leakage if query parameters contain sensitive information, for example via a phishing attack or just accidentally. Referrer-policies allow site owners to control what kind of user information is shared with external domains, but only 10% of sites deploy them.

Strict Transport Security (HSTS)

Most web users are familiar with the secure HTTPS connection that adds a layer of reassurance when they visit any website. HSTS is a security header that forces browsers to only accept secure HTTPS connections, ignoring any attempt to load resources from an insecure HTTP connection. As such, HSTS can protect users from protocol downgrading attacks (where an attacker forces a network channel to switch to a less secure standard), cookie hijacking and other man-in-the-middle attacks.



Few of the organizations that implement CSP have managed to do so in a meaningful, effective way.

Compared to CSP and SRI, HSTS is relatively easy to enable on a website; it's recommended that all website owners apply it but only 52.5% of the Alexa 1000 have done so. Again, it's a glass-half-full situation: that's 10% more than in 2019 but still means almost half of the world's leading websites aren't implementing even this low-hanging fruit on the security controls tree.

CONCLUSION

The stakes have never been higher

2020 has already been an unprecedented year for many reasons. As more consumers seek online services and e-commerce opportunities - and increasingly expect their personal data to be protected - the stakes have never been higher. At the most fundamental level, customers expect websites and web applications to be trustworthy. Many enterprises have invested significant resources in securing the network side of their business and customer data. They've invested significant resources in innovating the best possible user experience online. What they continue to overlook is the threat sitting in the shop window: client-side risk.

This year's Global Data at Risk - the State of the Web 2020 Report findings suggest that while high-profile attacks like Magecart have increased enterprise awareness of the vulnerabilities, and perhaps even driven more companies to at least try implementing controls like CSP, the road ahead is a long one. As this study suggests, effectiveness is heading in the wrong direction. Attacker effectiveness will not follow suit.

Want to learn more about Magecart attacks and client-side web and application security

Tala's blog, social media and website includes a lot of informative, awareness-raising and educational content. The biggest reason behind the success of Magecart-style attacks and customer data leakage is lack of awareness.

Enterprises interested in understanding their risk exposure to Magecart and other client-side vulnerabilities can visit www.talasecurity.io to book a comprehensive website risk assessment or demo to gain insight into how these threats impact your website and web applications.

About Tala Security

Tala's AI-powered website analysis engine was used to generate the data for this study. We make extensive use of AI and machine learning to collect the intelligence and data needed to identify website architectures, integrations, security status and exposure to client-side attacks like Magecart or third-party data leakage.

Tala's technology uniquely protects both websites and web applications. Our core capability is the dynamic deployment and continuous adjustment of standards-based security measures to deliver the comprehensive client-side security the modern web needs. Our AI-powered analytics engine evaluates over 150 unique indicators on a webpage's architecture and integrations, to build a 360-degree Application Information Model (AIM) and deliver application-aware security to all web assets. This comprehensive, continuous insight allows Tala to identify the optimal deployment and dynamic adjustment of browser-native, standards-based web security policies, including CSP, SRI, HSTS and other evolving web security standards. Tala's technology automates the policy generation process, including updating, implementation, alert analytics and incident management.

With Tala, a website can be running standards-based security in minutes, preventing website and web application attacks in real time, without impacting website performance.

Thank You!

Get in touch today

www.talasecurity.io
info@talasecurity.io