source DEFENSE

Out of Compliance and Out of Sight

The Client-Side Web Security Gap That's Putting Your Business at Major Risk

INTRODUCTION

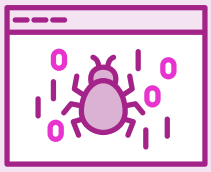
Client-side attacks on web applications — digital skimming, formjacking, clickjacking, ad injection, content defacement, and others — are some of the biggest threats today to organizations doing business online. These attacks take advantage of vulnerabilities in ubiquitously utilized first- and third-party JavaScript running on websites. Yet many organizations are unaware of the growing risk or believe that their current security measures are sufficient.

The simple and sobering fact is that almost every website — whether it's ecommerce, travel, healthcare, financial services, government, and so on — is vulnerable to the attack vector that enables these nearly invisible yet potentially devastating attacks. That's because 97.6% of web applications use JavaScript as their client-side programming language,[1] making these attacks one of your company's biggest cybersecurity gaps. This is particularly true for companies that collect information that requires compliance with privacy mandates such as the General Data Protection Rule (GDPR) and the California Consumer Privacy Act (CCPA).

Whether you're responsible for application development, application/cyber security, governance risk and compliance, or the digital line of business, client-side web application protection should be high on your priority list to proactively address. In fact, your business and your customers could already be victims of cybercriminals such as the Magecart syndicate that specializes in digital payment card theft.

In this white paper, you'll learn why every company and organization that conducts business online should be taking steps now to reduce the risk of successful client-side web application attacks, why Gartner recently named web app client-side protection as a soon-to-be mainstream security focus, and the best approach for protecting your brand and your customers.

1. "Usage Statistics of JavaScript as Client-Side Programming Language on Websites," W3Techs, October 2021



Magecart massively expands its scope

In November 2021, the National Cyber Security Centre (NCSC) announced that 4,151 retailers had been compromised by hackers attempting to steal customers' payment information and other personal data via client-side vulnerabilities on checkout pages.

In 2020, cybercriminals used the same techniques to compromise an estimated 2,800 retailers, injecting malicious code to steal the payment details of tens of thousands of customers. The attack is considered the work of Magecart, which uses JavaScript malware to target shopping carts associated with the Magento open-source e-commerce platform.

Sources: "Hackers Used This Software Flaw to Steal Credit Card Details From Thousands of Online Retailers," Danny Palmer, ZDNet, November 2021
"Cardbleed: 3% of Magento Install Base Hacked," Sensec, September 2020

QUANTIFYING THE RISK

Until recently, most organizations have vastly underestimated the risk that client-side web application attacks present. It's critical to understand that not only is it a major attack vector, but it's a growing one as well. Client-side attacks are one of the most lucrative and popular exploit techniques in use today.

In 2021, formjacking was responsible for 61% of web breaches.[2] Since 2017, 150 million payment cards were detected as being compromised via Magecart attacks, with cybercriminals attempting to monetize the cards on the dark web for an estimated total of $37 billion.[3]

With attention focused primarily on server-side security measures, numerous brands have been successfully attacked — including Macy's, Ticketmaster,

2. "2021 Application Protection Report," F5 Labs
3. "Rising Magecart Attacks Place Victims in Jeopardy," Christopher Thomas, About-Fraud, October 2021

American Cancer Society, P&G's First Aid Beauty, British Airways, and others — with serious consequences:

- British Airways was originally fined $238 million for GDPR violations that resulted from a Magecart attack and later agreed to pay a reduced fine of $26 million given the impact of the pandemic.[4]

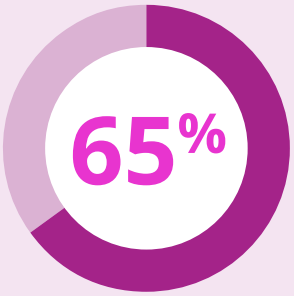
- Ticketmaster UK was fined $1.6 million under GDPR for a data breach stemming from third-party JavaScript code on its payment page, affecting nine million European customers.[5]

- Macy's was hit with a lawsuit over a Magecart data breach and the company's stock price took a 10% hit following the breach being made public.[6]

These companies aren't alone in the level of risk or potential impact of this type of attack. The vast majority of websites in the world today are susceptible because web app client-side protection is just now coming to the forefront of cybersecurity priorities.

Client-side risk also includes non-compliance not stemming from attacks, but from application programming mistakes. AllyBank disclosed that its website had sent customers' usernames and passwords in unencrypted text to external partners. The financial services company currently faces a class-action lawsuit that could have expensive consequences.[7]

4. "British Airways' GDPR Fine Dramatically Reduced," Doug Olenick, *Bank Info Security*, October 2020
5. "Ticketmaster UK Fined $1.6M under GDPR for 2018 Data Breach," Aaron Nicodemus, *Compliance Week*, November 2020
6. "Macy's Hit With Malware Attack, Customer Data Stolen," Paul Ausick, 24/7 Wall St., November 2019
7. "Ally Bank, Ally Financial Customer Info Breach Class Action," Class Actions Reporter, August 2021



Nearly two-thirds of consumers would leave if you had a data breach

65% of e-commerce shoppers say that "experiencing even a single data security breach would prompt them to leave a merchant for good."

Source: "Report: 65 Pct of Consumers Would Abandon Merchant After eCommerce Data Breach," PYMNTS.com, May 2021

Compliance fines represent one of the potentially devastating costs that organizations face when client-side attacks or mistakes occur. A data breach or leakage can also incur other direct costs such as legal fees, settlements of lawsuits, damages, forensic investigation, audit, and remediation. Indirect, but not insignificant, costs include brand reputation loss, customer churn, product delays, and downtime — all of which can impact revenue, company valuation, stock price, and shareholder value.

To help quantify the impact, today the average cost of a data breach is $4.24 million, up from $3.86 million in 2020. Of that total, detection and escalation accounts for $1.24 million, notification is $0.27 million, post-breach response is $1.14 million, and lost business represents $1.59 million or 38% of the total breach costs. The loss of customer personally identifiable information (PII) results in an average cost of $180 per lost or stolen record.[8]

UNDERSTANDING WHY CLIENT-SIDE WEB APPS ARE VULNERABLE

JavaScript vulnerabilities are not new, so why are client-side attacks not only still successful, but accelerating in volume and scope? The first reason is that until recently, there was a lack of awareness about client-side web app vulnerabilities, with many companies believing that existing, server-side security measures were enough.

8. "Cost of a Data Breach Report 2021," Ponemon Institute and IBM Security, July 2021

The larger reason, however, is the growing complexity of the digital supply chain. The average web application has between 40 and 70% of its code being sourced from third parties. These third-party scripts deliver a wide variety of rich functionality, including ads, analytics, social media, trackers, and much more, to enhance the customer experience as well as generate revenue. These third-party scripts often source additional content and functionality from fourth parties, further extending the web page supply chain.

Third-party and fourth-party scripts expand the security and compliance risk exponentially. That's because of a major security gap in JavaScript that gives all scripts — regardless of the source — the same level of control on the client side. Here's how it works:

1. Content is served and enriched: Web application logic — a combination of the owner's application logic and the integration of third-party content and functionality — is loaded and runs on the client side in the browser, beyond the protection of server-side security. The code is dynamically downloaded from a remote server, which means that it bypasses the traditional security infrastructure, including the website owner's firewalls and web application firewalls (WAFs).

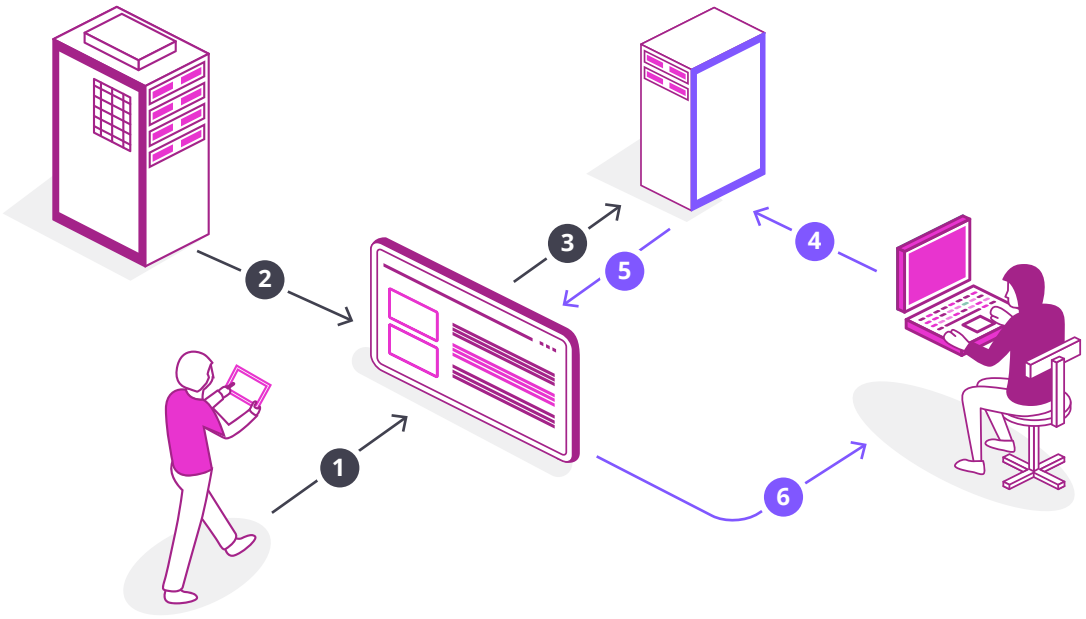
2. All scripts have the same level of control: Third-party and fourth-party scripts have the identical level of control as the website owner's own script. Every script on the page, no matter its origin, has access and authorship capability, meaning it can change the webpage, access all information on it (including forms), and can even record keystrokes and save them.

3. The vulnerability is easily exploited: No component of traditional security programs can prevent client-side attacks perpetrated via JavaScript. All it takes is for the third-party vendor to be hacked and have its code changed or an internal developer to integrate malicious code, whether accidentally or

intentionally. Website owners have limited means to dynamically detect the change and no means to prevent it from exfiltrating data or executing other malicious activity from the customer's browser.

Because it's difficult for website owners to detect, the majority of these attacks aren't discovered for weeks or months, increasing the scope of damage and mitigation costs and fines significantly.

Figure 1. Client-Side Web Application Attack



1. A user visits your website.
2. Content is presented in the user's browser from your corporate web server.
3. The content is enriched with third-party JavaScript.
4. An attacker compromises the third-party server.
5. Malicious content is served to your website visitor's browser.
6. The attack is successful. Customer data is exfiltrated, putting company revenue and brand integrity at risk.

ASSIGNING RESPONSIBILITY FOR PROTECTING AGAINST CLIENT-SIDE ATTACKS

Unlike other types of attacks on end user devices such as phishing or ransomware, the responsibility for client-side web app protection falls squarely on the website owner. Courts and governmental authorities are holding businesses accountable for theft or leakage of customer PII, even if the malicious code came from a third-party vendor. The GDPR goes as far as specifying liability to the website owner for third-party behavior.

This means that brands and organizations must take steps to protect their customers' data — and their company's reputation, shareholder value, and revenue — from client-side attacks. They cannot rely solely on third-party or fourth-party vendors to detect and prevent malicious code from being downloaded from their servers to the customer's browser.

While the accountability for data theft or leakage lies with the company, determining who within the company is responsible for ensuring protection of client-side web applications often spans multiple stakeholder groups. This makes it more complex and challenging to agree on and quickly implement an approach to mitigating the risk. Stakeholders involved often include:

- **Website owner/line of business:** Many of the third-party scripts used to deliver client-side functionality are included because the business wants to improve the customer experience and/or generate additional revenue streams. Any measures taken to protect the web application from client-side attacks must not degrade the customer experience or negatively impact business outcomes.

- **Web development team:** As web application complexity grows, developers rely more and more on open source components to reduce the JavaScript

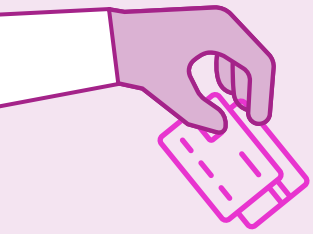
programming effort. Additional testing processes over and above the validation testing conducted as part of a CI/CD pipeline could create friction that slows down development and the release of new functionality.

- **Cyber/Web/IT security team:** Security solutions that generate significant amounts of alerts can overwhelm an already heavily burdened security team, causing client-side web application protection to be prioritized lower than existing security tasks.

- **Governance, risk, and compliance (GRC) team:** Tasked with making sure the company is compliant with consumer privacy regulations, these teams may not be aware of dynamically loaded scripts and content from third and fourth parties, and the security vulnerabilities this creates. The GRC team needs a solution that will help the company pass audits and reduce the risk of non-compliance.

- **Finance and legal teams:** Reducing compliance and legal risk are important to stakeholders within the finance and legal departments as well. They will want to err on the side of caution to protect the company from attacks or data leakage that could affect the company's financial health and legal liabilities.

It's critical for all of these stakeholders to be educated on the level of risk — including potential impact on cash flow, customer trust and loyalty, brand reputation, stakeholder value, credit risk, and more — that client-side attacks represent for the company so they can make protection a high priority.



50,000 runners get their cards stolen

More than 50,000 compromised Card Not Present payment records were stolen from Running Warehouse and then offered for sale on the dark web between October 2020 and October 2021.

Source: Gemini Advisory, October 2021

CLOSING THE JAVASCRIPT SECURITY GAP

We already know that traditional server-side security doesn't address the risks discussed because client-side scripts operate completely outside of the security capabilities an organization deploys to secure the server side of the browser session.

However, what about other security measures that may already be in place, such as application security validation testing or dynamic application security testing (including *AST testing (IAST, RASP, SAST, DAST, static code analysis, etc.)? While these testing processes evaluate the integrity of the designed JavaScript call function, they are not designed to test every use case or operate dynamically, nor can they test the code residing on a third-party or fourth-party remote server. They are also not capable of providing real-time scanning of all web traffic across the entire user population.

Likewise, using content security policy (CSP) and/or subresource integrity (SRI) features are not enough to protect client-side web applications from today's threats. While CSP and SRI can be powerful tools for website protection and data management, they have significant limitations that impact the ability for website owners to use these measures effectively against the top client-side threats:

- **Limited threat protection:** Neither CSP nor SRI-based security can protect against key logging attacks, sensitive data screenshots, formjacking, cookie takeover, or clickjacking.

- **Time to market delay:** Using CSP means that third-party scripts have to go through a cumbersome process of identification and whitelisting of actions, which affects the time to production for any third-party provider and requires developer resources.

- **Possible loss of functionality:** Because domains are whitelisted, any change to the internal domains for a third party (for example, a new content delivery network (CDN)) might cause the script to stop working. Identifying the cause of these issues might take days or even weeks.

- **Partnership impact:** Many third-party vendors rely on additional vendors to enhance their efficiency and revenue. Because fourth-party usage is dynamic, triggered by different user profiles and other factors, it makes these domains difficult to predict in advance so they can be whitelisted for CSP. The result is that many fourth-party partnerships could be blocked.

There are tools entering the market that can monitor and alert on suspected malicious code in JavaScript running on the client side. They are indeed part of the solution for addressing client-side web app threats.

That said, focusing only on detection has a few drawbacks. First, the website owner must investigate each alert, determine whether it represents a true threat, and remove the malicious code. Second, monitoring-only tools can generate thousands of alerts in a short span of time, with many of them likely to be false positives that must be investigated by the cybersecurity team. Signal fatigue and operational burden can become a major issue as a result. By far, the greatest drawback is that detection-only solutions don't prevent an attack from being successful.

STOPPING THE PROBLEM BEFORE IT HAPPENS:

Detection, Protection, and Prevention

The best approach to client-side web app protection is a prevention-first solution that's purpose-built to detect, protect, and prevent client-side web app attacks in a way that does not:

- Add operational burden for the cybersecurity team
- Slow time to market with new third-party capabilities or content
- Generate thousands of false positives that must be investigated
- Negatively impact website performance or customer experience

The Source Defense patented Website Client-Side Security Platform offers the most comprehensive solution to detect website skimming, formjacking, and supply chain attacks and stop them before they affect your website or your customers. Source Defense reduces risks and vulnerabilities while providing an optimal user experience. It's also architected for deployment and administration simplicity. On average, Source Defense users spend less than five hours per month managing the solution on their production websites.

Source Defense uses real-time, client-side sandboxing and permissions-based isolation and reflection to protect your company and your customers' data and prevent successful data exfiltration or leakage by:

- Isolating and monitoring JavaScript execution in an end user's browser, in real time, as the user interacts with your web page
- Using real-time JavaScript sandboxing to restrict the access that each script has to a web page as well as control that script's behavior
- Allowing or restricting access to different parts of the page and the data that they contain
- Monitoring and managing the flow of data from the page to other places
- Enforcing security controls

As far as things that are easy wins in information security, as few as they are, Source Defense VICE is a gem. From onboarding to normalized operations, the Source Defense team moves quickly and efficiently to protect customer sites. Customers have a clear and simple dashboard to utilize although you won't have much reason to logon. The Source Defense team is constantly monitoring and adapting the system to provide you with the best security service. The lack of alerts to you from Source Defense is a testament to the efficiency of the ML and the Source Defense human intelligence team. Learn a new way to spend your nights and weekends — relaxing. So easy, works so well, unconscionable for an information security professional to not have Source Defense VICE in place."

— A multibillion-dollar global sports equipment and entertainment company

CONCLUSION

Cybercriminals like those in the Magecart syndicate are increasingly targeting unprotected web applications on the client side because the JavaScript security gap highlighted here is an opportunity too lucrative for them to ignore.

That's why you need to protect your business and your clients from becoming the next victims with a solution that detects and prevents client-side attacks from being successful. Source Defense offers the only purpose-built, patented technology for real-time protection against risks and threats originating in JavaScript.

To learn more, visit
https://sourcedefense.com/request-a-demo/

To see how your website stacks up against threats and understand your risk, request a free risk report by visiting
https://sourcedefense.com/check-your-exposure/

source
DEFENSE

About Source Defense

Source Defense is the market leader in Client-side Security for websites, providing real-time threat detection, protection and prevention of vulnerabilities originating in JavaScript. The Source Defense patented Website Client-Side Security Platform offers the most comprehensive and complete solution to address threats and risks originating from the increased use of JavaScript, third-party vendors, and open source code in websites today.

The ADMIN management console, VICE sandboxing and WiPP data shield offerings utilize patented technology and are deployed by leading Fortune 500 enterprises in the financial, retail, eCommerce, and healthcare markets. Headquartered in Israel with branches across the U.S. and a strong community of global valuable partnerships, Source Defense is the most innovative, reliable, and trusted partner in the fight against client-side attacks.