Lack of awareness is a key driver for the acceleration of client-side attacks. Tala Security CTO Swapnil Bhalode analyzes the key attack vectors - and shows how they can be resolved.

More than 90% of websites globally rely on external content that’s either written or managed by third parties - of a median 73 resources, 23 are loaded from external domains. When you consider that a single XSS vulnerability compromises the entire domain on which it occurs, that’s a lot of vulnerability for your business in the hands of someone else’s: Magecart, user data leakage, content integrity attacks, ad injections, session re-directs, and cart hijacking all exploit these vulnerabilities, often with serious business consequences.

That's the bad news. The good news is there's something you can do about it. Better still, it’s not complex and it won’t impact your website's performance.

And it starts with a little insight into how these major attacks work – and how Tala detects and protects against them.

**Cross-site Scripting / XSS**

XSS features consistently among the Top 3 vulnerabilities detected on websites. There’s no wonder cybercriminals and other fraudsters love it so much: not only is it a widely available attack surface, but a single XSS vulnerability compromises the domain on which it occurs.

- **XSS: REFLECTED**: Reflected XSS is a non-persistent XSS attack, where a malicious script is reflected back to the user by the end server. The attack is typically exploited via a link that is sent to the victim. This link embeds a malicious script, which is not sanitized by the end server, and is sent back to the end user’s browser session where it executes.

- **XSS: PERSISTENT**: This where the malicious script gets stored on the server side, usually in a DB, as legitimate content. The attack typically takes place on web app areas where users submit input, through which the malicious script is injected. With Persistent XSS, the end server stores the malicious script. Because of this, the attack executes whenever the script is fetched from the server.
Tala detects and prevents both Reflected and Persistent XSS attacks by analyzing the app, creating a list of all legitimate scripts (inline and non-inline), and whitelisting them through CSP’s ‘script-src’ directive. The reflected malicious script will not be allowed to run because it won’t be part of the whitelist. Tala uses nonces to protect against malicious inline JavaScript or tampering. Any inline JavaScript that Tala determines is malicious will not be certified with a nonce.

- **XSS DOM**: Malicious script is executed on the client as a result of modification of the DOM dynamically. Unlike Reflected and Persistent XSS, there’s no server-side vulnerability with DOM XSS. The attack typically is delivered via links where an input element in the link is modified to include the malicious script. This input becomes part of dynamic code execution on the client side, and gets executed.

Tala detects all the DOM XSS sinks an application might be using (which lead to a DOM XSS attack). CSP is currently limited in its ability to protect against DOM XSS but Tala is exploring a forward-looking solution proposed by W3C.

**Data Exfiltration**

In this attack, user data is stolen and sent to the attacker’s server. There are a variety of ways this attack could take place: for example, a form injected by the attacker to lure the victim into submitting their credentials. Or it could be a Magecart-style attack, where a 3rd party JS is compromised to steal user data (such as payment information).
Types and examples of Data Exfiltration attacks

3RD PARTY COMPROMISE (MAGECART):
Third-parties could be compromised in a number ways, outside the scope of the web application they’re integrated into. Magecart is one of the largest groups of cyber-criminals targeting this client-side vulnerability in enterprise websites – security researchers have held Magecart responsible for attacks on websites belonging to British Airways, Ticketmaster, NewEgg, OXO and thousands of other enterprises.

WHAT’S AT STAKE IS SIGNIFICANT, AS CONTINUED ATTACKS RISK THE EROSION OF THE MOST IMPORTANT INGREDIENT THAT POWERS E-COMMERCE: TRUST.

Magecart primarily launches attacks by adding “card skimming” code into legitimate JavaScript files served on a website. When a user visits the site and types sensitive data such as credit card numbers, the “card skimming” code sniffs the information via the browser and sends it to a malicious server. These attacks include first and third-party JavaScript/supply chain compromises, cross-site scripting (XSS), ad injections and other forms of client-side attacks.

The acceleration of successful Magecart attacks has exposed fundamental and universal vulnerabilities in web security and served as a troublesome indicator of the lack of effective defenses for combating this growing threat. What’s at stake is significant as continued attacks risk the erosion of the most important ingredient that powers e-commerce: trust.

PII DATA EXFILTRATION
Web apps trust and integrate with 3rd party services, e.g. analytics, user tracking etc. These services can potentially, and inadvertently, access sensitive user information and send it out to their own servers. This could lead to loss of trust on the end users’ part, along with compliance-related issues.

Tala uses fine-grained CSP policies, as well as SRI (integrity hashes), and continuous monitoring to detect these attacks during scanning.
Content Injection Attacks

UNAUTHORIZED CONTENT INJECTION
Content injection is a generic attack where malicious HTML content is added to the web page. Content can be injected via a variety of vehicles, such as malicious extensions, XSS, etc.

```
<html>
  <tag/>
  <code/>
</html>
```

- BELONGS TO THE WEB PAGE
- UNAUTHORIZED INJECTED CONTENT
  (XSS, 3RD PARTY COMPROMISE, BROWSER EXTENSION)
Types of Content Injection Attack

COMPETITOR AD INJECTION
Malicious or competitive, non-approved ads placed in the browser, e.g. this mocked-up screenshot of a website that the end users are intentionally browsing to (nordstromrack.com) but where they're seeing ads for a competitor (macys.com):

MALICIOUS IFRAME INJECTION
An iFrame is injected into a web page, the content of which is loaded from a malicious website. The malicious websites typically contain exploit code that can potentially compromise the end user’s machine. The attack is typically constructed by exploiting a server-side vulnerability. The application’s code is modified to include the malicious iFrame. iFrames can also redirect users to malicious websites.

Tala protects against Content Injection attacks by using all the directives supported by CSP (img-src, font-src, style-src etc) to prevent any code or markup that is injected. Tala also uses SRI to prevent any 3rd party code modifications that could lead to content injection attacks.
CLICKJACKING ATTACKS
Clickjacking is a ‘UI redressing attack’ where multiple layers of content are created on top of each other and hidden in order to trick the user into doing something unintentional. The attack is typically hosted on a website created by the attacker and users are lured there by clicking on a seemingly relevant or interesting link, for example a banking-related one. The malicious website is specially crafted to show fake content, hiding the real content, which is hosted through ‘iFrame’ - e.g. a login page to the victim’s bank account). Here’s how it looks:

Tala protects against Clickjacking attacks using the ‘frame-ancestors’ directive provided by CSP. This ensures that the customer’s website can only be embedded on whitelisted domains.

MALICIOUS/SIDELOADED BROWSER EXTENSIONS
Malicious browser extensions can perform various activities such as Ad Injections, data theft etc. Malicious extensions are typically loaded by malware running on the end user’s machine.

Tala detects and prevents malicious browser extension attacks by using all the directives supported by CSP to prevent any code or markup being injected by the extension.
You Don’t Have to Sacrifice Performance for security

Only 2% of website operators deploy CSPs capable of preventing client-side attacks. Activating standards-based security ensures exceptionally efficient website performance. By using the standards that are already in place, already browser-native, you get all the control with no additional overhead. When you automate that process, as Tala does, you can achieve unmatched performance without compromising on client-side security.