

# Radware Client-Side Protection



While enterprises do their best to protect customers' personal data on their application environments, the information end-users enter on their client side (for example, ID numbers, address, credit card number, contact information and so on) can be exposed to third-party services embedded in the applications—which are automatically trusted by the main application, but rarely monitored. An average application runs **dozens of different third-party JavaScript services** (Outbrain, Google Analytics, Tranzila and so on) that are loaded when the user first visits a page. These services are often referred to as the application supply chain. Some of those services are even dependent on their own supply chain with services from 4th and 5th parties.

These services are loaded when the user first visits a page, while an unmonitored legitimate connection is established between the end user and the relevant third-party servers. Basic security tools cannot monitor this data path and are blind to any malicious code that might be coming to the client side from the third-party server.

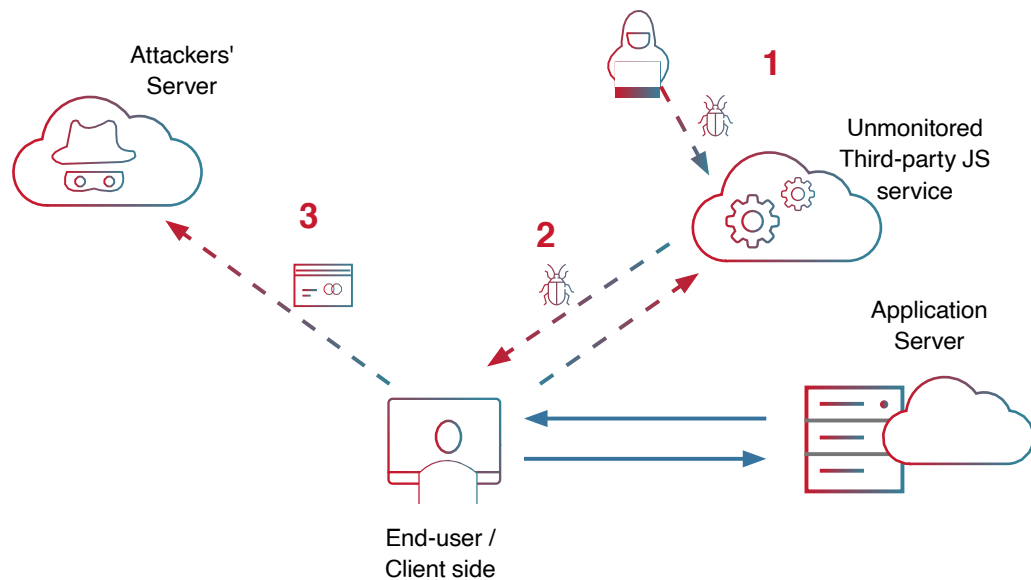
## Reasons for Activating Client-Side Protection

As server-side security improves, more hackers target the less protected and rarely monitored client side. Client-side protection needs to be activated primarily to handle the following challenges:

- **Increase in malicious supply-chain attacks through third-party services JavaScript** (Magecart, skimming, Formjacking)
- **Lack of Visibility and Control over third-party services –**
  - Unable to detect if the JS code of services in the supply chain has been breached or tampered with
  - No control of third-party services' security
  - No monitoring of supply chain (sub-services from 4th party, 5th party, and so on)
- **Compliance and liability –** As organizations are responsible for the safety of their end-users data and PII, they need to ensure that their customers' privacy is not being jeopardized by any third-party services incorporated into their applications.

**Figure 1**

Magecart/ Skimming/ Formjacking Attacks – completely invisible to the end-user. Sophisticated attackers breach a server on the supply chain and hide malicious malware that injects a script directly onto targets' forms, which then collects the sensitive information that the end-user enters and sends it back to the attackers' remote server.







1. The attacker hacks a server on the supply chain and hides malicious malware
2. The infected server returns a response, injecting malicious script into the client's form (or added fake input fields)
3. The attacker collects and sends the user information to his remote server.

## Radware Client-Side Protection Solution

As part of Radware’s One-Stop-Shop Application Protection service that protects the application data center and functionality, this solution offers advanced Client-Side Protection that ensures the protection of end users' data when interacting with any third-party services in the application supply chain.

- **Protect end-users from client-side attacks coming from third-party JS services.** (Formjacking, Skimming/Magecart)
- **Visibility** – Discover, map, and assess third-party JavaScript-based services embedded in the application.
- **Easily block requests to suspicious third-party services in the supply chain.**
- **Adhere to data security compliance standards.**

## Advantages of Radware Client-Side Protection

Visibility	Client-Side Attack Protection
 <ul style="list-style-type: none"> <li>➤ Continuous discovery of all 3rd party services in your supply chain</li> <li>➤ Detailed activity tracking</li> <li>➤ Alerts &amp; threat level assessment according to multiple indicators including script source and destination domain</li> </ul>	 <ul style="list-style-type: none"> <li>➤ Magecart and various skimming attacks</li> <li>➤ Formjacking attacks</li> <li>➤ Supply chain exploits</li> </ul>
Data Leakage Prevention	Surgical Enforcement
 <ul style="list-style-type: none"> <li>➤ Unknown destinations</li> <li>➤ Legitimate destination with illegitimate parameter</li> <li>➤ DOM Based XSS</li> </ul>	 <ul style="list-style-type: none"> <li>➤ Not standing in the way of vital JS services</li> <li>➤ Blocking only nefarious scripts</li> </ul>

**Complete End-To-End Protection**  
**Auto Discovery - > Risk Assessment - > Mitigation**

