Protect Your Website From Magecart During the Holidays

Magecart checklist

Magecart Web Skimmers

“Magecart” refers to a collective of cybercriminal groups that **inject digital credit card skimmers** on e-commerce and payment websites. These groups have been active since 2015, but have gained momentum from 2018 onwards.

In a Magecart attack, attackers inject the skimmer (through malicious JavaScript code) into a company’s payment page. This code collects credit card details whenever a user submits them in a form (formjacking) and sends them to attacker-controlled drop servers. During this process, **neither the end-user nor the company have any awareness that the attack took place.**

Attackers may gain access to the victim’s website and directly place the skimmer in the payment page (first-party attack) or **inject the malicious code through a third-party provider** that the victim company is using (such as a live chat tool, analytics service, or code dependency). These third-party Magecart attacks are critical because they don’t require a first-party server breach or direct access to the company’s website. Instead, they target companies’ third-parties, which often have **fewer resources dedicated to security.** And this (infected) third-party code has all the same permissions as all the other code in a website.

Magecart will certainly get coal in its stocking this year:

$1.3B+
business losses from known Magecart attacks.

$256M
GDPR + ICO fine on British Airways following a Magecart attack

22 days
average time during which Magecart skimmers remain active before discovery

Magecart Mitigation Checklist

Because Magecart Mitigation is a complex topic and mainstream security solutions aren't capable of preventing these attacks, it's important that companies know how to properly assess a security product to mitigate web skimmers. So, this checklist recommends **technical tests** that should be performed when testing such a product, as well as **important technical requirements** to consider when procuring a vendor.

1 - Technical Tests

Action

1.1 Detect and block the addition of "click" or "submit" event handlers to the page. The addition of form-related event handlers (for example, of an onmouseover event) is a common malicious behavior in Magecart skimmers.

1.2 Detect and block the addition of elements to the page, such as forms. More advanced web skimmers add fake credit card payment forms to the page or new buttons to the page. This sort of document object model (DOM) tampering is a common indicator of malicious behavior.

1.3 Detect and block the removal of elements from the page, such as a div and its child nodes. By removing content from the page, attackers can divert users from the legitimate flows in lieu of compromised ones.

1.4 Detect and block the modification of page content, such as editing element attributes or changing element visibility. Much like removing elements from the page, modifying it lets attackers trick users, for example by hiding a spinner.

1.5 Detect and block sensitive data collection and its exfiltration. Magecart attackers invariably need to send the captured data out to a drop server. Security teams need to detect this, namely by monitoring for outbound network events to unknown domains or even unexpected data to known domains.

2 - Important Technical Requirements

Besides the technical tests outlined in the previous topic, there are some key features and important red flags to be aware of when procuring a Magecart Mitigation product

Action

2.1 Require a complete website inventory. This improves visibility of the scripts and network connections that take place in any given user session, making it easier to learn what's normal and to spot malicious behaviors.

2.2 Avoid bot-based approaches. Some of the more advanced Magecart skimmers use bot detection techniques to avoid detection from approaches that visit the page continuously to check for skimmers.

2.3 Avoid products with limited compatibility. Some products don't work on all browsers and versions; for example, SRI isn't compatible with Internet Explorer or with Safari for iOS.

2.4 Avoid any type of impact on page performance. Online experts consider page performance an important driver of e-commerce sales; the solution should leave a minimal footprint in performance.

2.5 Avoid high-maintenance products that are difficult to integrate. Integrating a product that requires significant refactoring of current systems or substantial maintenance and configuration effort will lead to problems further down the road.

2.6 Avoid approaches that are only signature-based. These products will be very limited in terms of detection capabilities and will likely fail to detect new exploits. Optimal products look for behaviors instead.

2.7 Require tamper-resistant defense code. The defense code will often run alongside potentially malicious code. To avoid interference from the malicious code, the code itself must deter direct tampering attacks.

Frequently Asked Questions

Doesn’t a web application firewall (WAF) prevent Magecart attacks?

No. While WAFs block unknown or untrusted network connections, they don’t detect what happens on the client-side. Since Magecart attacks originate from a source that is trusted by default (a legitimate third-party supplier or a piece of first-party code), the malicious web skimming code easily bypasses WAFs.

Don’t CSP and SRI prevent Magecart?

In short, no. Even if companies successfully set up CSP and SRI (which are tricky to get right), there are known bypasses and many other pitfalls that mean that these strategies can’t guarantee enough protection against web skimmers.

If a website is developed and maintained in-house, isn’t it “immune” to Magecart?

No. Even if everything is self-hosted, with minimal reliance on third-party code, first-party Magecart attacks can still happen and go undetected for weeks.

Can I test Jscrambler’s magecart mitigation product before committing?

Yes. We can provide **3-month long detection of Magecart web skimmers** with our Magecart Mitigation product. This includes **periodic reports** of detected malicious behavior and suspected Magecart attacks, a **complete website inventory** (with details of infected scripts and suspicious network events), and a **thorough analysis of each detection** (including its source and call stack).

How A Major E-Commerce Company is Mitigating Magecart Attacks with Jscrambler

Overview

We were approached by a major airline wanting to **mitigate Magecart** web skimmers and **prevent them from running undetected** on their websites **and exfiltrating data.**

By employing Jscrambler Webpage Integrity, the airline started detecting malicious behaviors on the client-side, analyzing them with a **comprehensive inventory** and live feed, and using our powerful rules engine to block Magecart web skimming attempts.

Webpage Integrity met every strict test and requirement and was smoothly deployed to production, with a neglectable performance impact. The airline now continually monitors for malicious client-side behaviors, not limited to Magecart but other client-side threats. Thanks to easy configuration and maintenance, their security team can quickly adjust their rules and meet new security challenges head-on.

Challenges:

- Mitigate Magecart credit card skimmers;
- Reduce exposure to data leaks;
- Integrate with their SIEM;
- Ensure minimal footprint.

Solution:

Jscrambler Webpage Integrity:

- Behavior-based detection
- Fine-grained behavior control;
- Comprehensive Inventory, detection live feed, and explorer;
- Real-time Magecart mitigation.

Results:

- Passed all detection tests
- Clear, accurate reporting of malicious client-side behavior;
- Much superior results compared to other tested solutions;
- Implementation passed strict requirements;
- Minimal performance impact;
- Smooth transition from PoC to a production environment;
- Easy configuration and maintenance.

If you want to know more about how Jscrambler can help you prevent client-side attacks, don’t hesitate to contact us.

hello@jscrambler.com | +1 650 999 0010