The Optimizely attack: when trusted third-party JavaScript turns hostile

17 November 2026 · 6 min read · blogs.iictp.org/optimizely-supply-chain-attack-third-party-javascript

Security researcher Scott Helme of Report URI reports that on 16 November 2026 malicious JavaScript was served to visitors of 178 organisations through the Optimizely Web experimentation platform. The affected sites include widely visited names in media, retail, software and travel.

TL;DR The short version

- Report URI says malicious JavaScript reached visitors of 178 organisations through the Optimizely Web platform on 16 November 2026.
- The code swapped crypto wallet addresses, skimmed card details, collected cookies and screenshots, and tried iOS exploits on older devices.
- Inventory every third-party script and enforce a Content Security Policy, because a site can be compromised without its own servers being breached.

Read online, listen to it, or change the text size: https://blogs.iictp.org/optimizely-supply-chain-attack-third-party-javascript

What was reported

- Optimizely Web works by having each customer load a JavaScript “snippet” from Optimizely’s CDN. The snippet contains an optional “Project JavaScript” field, custom code that runs on every page, according to Report URI.
- Report URI says the attacker wrote code into that field in customer projects. The file was still served from Optimizely’s trusted domain with its certificate, so sites that trusted that host loaded the malicious code.
- Report URI found 188 infected snippets across 178 organisations among 1,611 it checked, and says the real number is likely higher. It says most of the malicious code had been removed by the evening of the same day.
- Report URI says it does not know how the attacker gained write access to so many customer projects, and that this is for Optimizely to explain. It stresses that the CDN served exactly what it was given; this was not cache poisoning.

What the code did

Report URI describes a multi-purpose kit: it swapped cryptocurrency wallet addresses for the attacker’s, skimmed payment-card details typed into a site’s own form fields, collected non-HttpOnly cookies and page screenshots, and on older iPhones and iPads tried to load iOS exploit kits. Report URI says cards entered in hosted payment iframes were out of reach of the script.

What a practitioner should do

- Inventory every third-party script your pages load, including “marketing” and experimentation tags. Each one runs with the full power of your page.
- Deploy a Content Security Policy and move it from report-only to enforcement, so unexpected hosts and inline script are blocked. Report URI argues this could have blocked parts of this attack.
- Monitor script integrity: alert when a trusted script’s content changes, and when a page loads a script or connects to a host it never did before.
- Keep payment entry in hosted iframes from your payment provider, and mark session cookies HttpOnly.
- Treat any tool that lets staff paste custom JavaScript into production as privileged: restrict who can edit it, log changes, and review them.

The wider lesson

A site can be compromised through a dependency without its own code or servers ever being breached. Trust in a third-party host is a decision with a blast radius, and it needs the same review as any architecture choice. This is the reasoning the Infrastructure & Security domain of the TPBOK asks of a practitioner.

> Facts in this post were read from the sources listed below on 17 November 2026; this is a developing story or product and details may have changed since.

Sources and further reading

1. Report URI (Scott Helme). World’s Largest Brands Infected With Malware Via Optimizely. 16 November 2026. blog.report-uri.com/worlds-largest-brands-infected-with-malware-via-optimizely/

Published by IICTP, an independent technical credential body, on 17 November 2026. The current version of this post is at https://blogs.iictp.org/optimizely-supply-chain-attack-third-party-javascript. More guides: blogs.iictp.org.