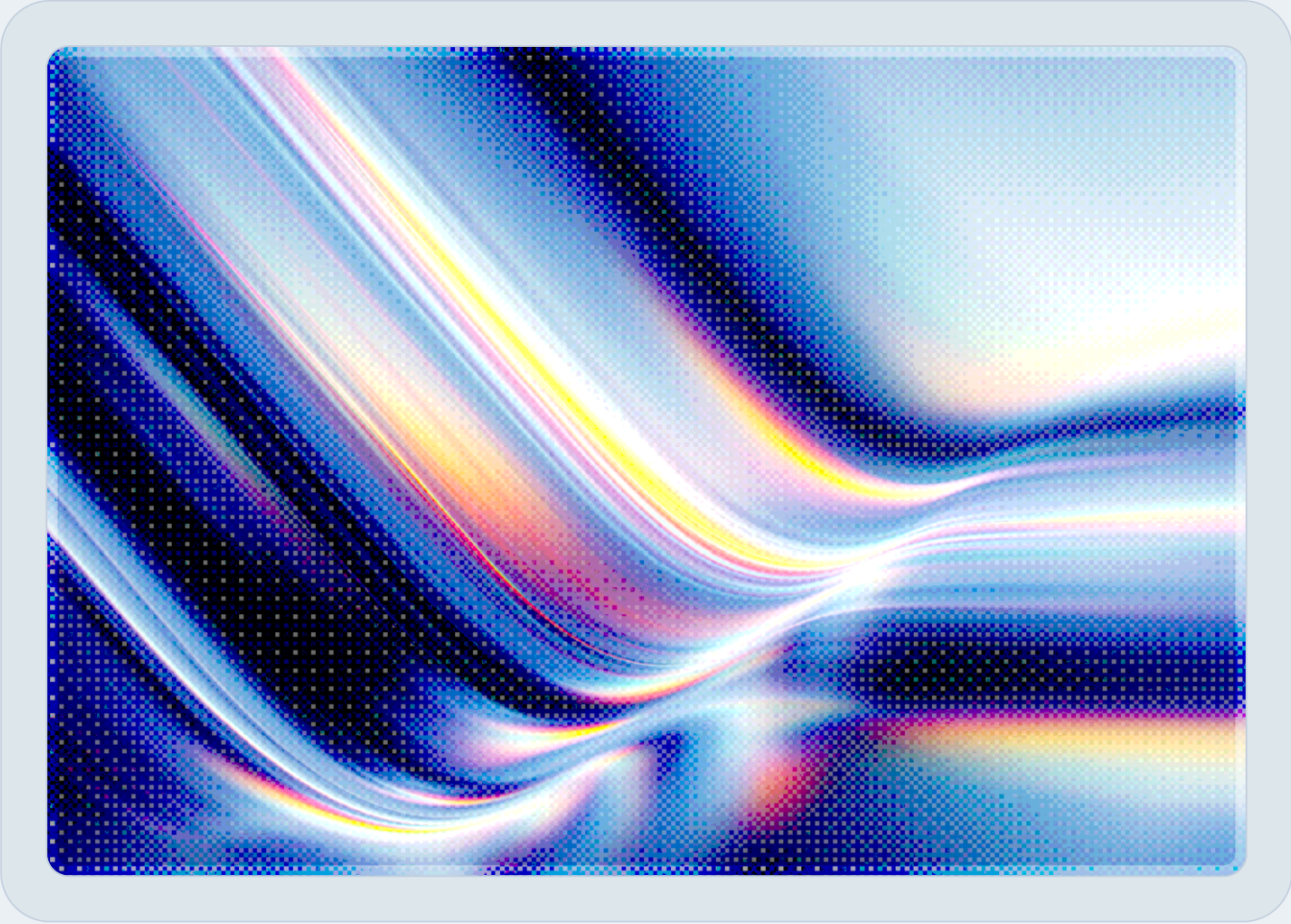
CSAI Foundation | Cloud Security Alliance

Brevo Breach: Stolen API Key Poisons 100,000+ Websites

2026-09-21

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

A stolen, long-lived Cloudflare API key with full account permissions — hardcoded in Brevo's application source code — let attackers create a malicious Cloudflare Worker that rewrote CDN edge responses without touching a single origin server, evading standard file-integrity monitoring [1][3]. Over a roughly four-and-a-half-hour window on September 14, 2026, the injected code reached more than 100,000 websites that embed Brevo's tracking, chat, or form widgets, delivering two distinct payloads: a ClickFix social-engineering overlay that tricked ordinary site visitors into pasting and executing Windows commands, and a self-hiding WordPress plugin backdoor deployed against administrators logged into their own dashboards [1][2][3]. Security research firm Sansec identified and disclosed the attack after tracing altered JavaScript files back to a shared malicious infrastructure cluster; Brevo revoked the exposed credentials, removed the rogue Worker and DNS records, and purged edge caches within hours of confirmation [1][3].

The incident occurred four days after a separate breach of Brevo's SAML single sign-on handling exposed roughly 138 customer accounts, including one belonging to hardware wallet maker Trezor. Brevo has not confirmed a causal link between the two incidents, but the proximity is a reminder that a single vendor's security posture can degrade across multiple, independently exploitable failure points in a short window [4]. Organizations that embed any third-party JavaScript, tracking pixel, or chat widget should treat this incident as a reminder that vendor-side credential hygiene is now part of their own attack surface, regardless of how the vendor's platform is configured on the customer side.

Background

Brevo (formerly Sendinblue) is a customer engagement and email marketing platform whose JavaScript SDK, live-chat widget, and web forms are embedded directly on customer websites, a deployment pattern widely used across the marketing-technology sector and one that gives a single vendor's compromised asset direct code-execution reach into every site that loads it. On September 14, 2026, between approximately 16:05 and 20:30 UTC, attackers used a stolen Cloudflare API key to deploy a malicious Cloudflare Worker across Brevo's content delivery infrastructure, altering the JavaScript files `sdk-loader.js` and `brevo-conversations.js` that tens of thousands of customer sites

load on every page view [1][2][3]. Because the modification happened at the CDN edge rather than on Brevo's origin servers, file hashes and deployment logs on the origin side showed nothing out of the ordinary, a technique that let the injected code run for hours before detection [1].

According to Brevo's own account of the root cause, the attackers obtained a Cloudflare API key with full account permissions that had been hardcoded directly into application source code, rather than stored in a secrets manager or scoped to a narrow set of operations [1]. Forensic analysis published by Sansec found that an SSL certificate for one of the attacker-controlled domains, `cdn.sendibt1.com`, had been issued on August 25, 2026 — roughly three weeks before the injection began — suggesting the infrastructure was staged well in advance of the active exploitation window, though Sansec found no evidence of malicious activity prior to September 14 [3]. Independent reporting has raised the possibility that the Cloudflare account compromise was connected to a separate, earlier incident in which Brevo identified and closed a flaw in its SAML single sign-on handling on September 10, 2026, after a threat actor used it to access approximately 138 customer accounts, including one belonging to crypto hardware wallet manufacturer Trezor [4]. Brevo has not confirmed a direct causal link between the two incidents, and this connection should be treated as an open question rather than an established fact.

Sansec's forensics team, escalated by initial alerts from web-security firm ParadoxLabs, identified the scope of the campaign and published a detailed technical analysis on September 16, 2026, two days after the injection stopped [3]. All attacker-controlled hostnames — a cluster of subdomains under `sendibt1.com` along with several unrelated-looking domains such as `yelahaye.surf`, `boiseno.club`, `glegchner.com`, and `corralos.beer` — stopped resolving on September 15, consistent with the attackers abandoning infrastructure once the compromise was detected and remediated [1]. Brevo responded by removing the malicious Worker and its associated routes, revoking the compromised API key, purging edge caches, and eliminating hardcoded credentials from its source code, though the company has not published a comprehensive post-incident report detailing the full remediation timeline [1][3].

Security Analysis

The technical core of this incident is a familiar failure pattern rather than a novel exploit: a single, overprivileged, long-lived credential embedded in source code gave attackers the same edge-level control over Brevo's CDN that Brevo's own engineers use in normal operations — a class of failure that credential-scoping and secrets-management practices are specifically designed to prevent. A Cloudflare API key scoped to full account permissions can create Workers, modify DNS records, and establish new routes — capabilities that, in the wrong hands, function as a content-management system for every domain behind that account. Because the attackers modified responses at the CDN layer rather than

altering files on Brevo's origin servers, conventional integrity checks that compare deployed code against a known-good origin state would not have flagged anything unusual, and the Content-Security-Policy headers that might otherwise have blocked the injected script were themselves stripped by the malicious Worker before the response reached the browser [1]. This edge-rewriting technique differs from more familiar third-party script compromises in one important respect: rather than altering a script file that customers then have to re-fetch, the attacker changes what every browser receives in real time, for as long as the Worker remains active — arguably a more severe failure mode because it requires no re-fetch or cache expiry to take effect.

Once in place, the injected code appended a short loader script to Brevo's legitimate JavaScript assets, which in turn fetched a second-stage payload from attacker infrastructure and executed different logic depending on the visitor [3]. Sansec's technical analysis identified command-and-control paths used for visitor fingerprinting, a proof-of-work challenge (likely intended to slow down automated security scanners), and staged command delivery, alongside a separate beaconing endpoint disguised as an image request [3]. The two customer-facing payload paths differed sharply in their targeting and mechanism, as summarized below.

Dimension	ClickFix Overlay (Site Visitors)	WordPress Plugin Backdoor (Administrators)
Target	Any visitor to an affected website	WordPress admins with an active `wp-admin` session
Delivery mechanism	Full-page fake "human verification" overlay mimicking a Cloudflare challenge	Silent plugin installation triggered when an admin loaded their dashboard
Social engineering	Instructs the visitor to copy and paste a command into the Windows Run dialog or terminal	None required — installation occurs via the admin's authenticated session
Persistence	None (relies on the visitor executing the pasted command)	Plugin hides from the WordPress plugin list and copies itself into the must-use plugin directory, which loads automatically and is harder to disable

Dimension	ClickFix Overlay (Site Visitors)	WordPress Plugin Backdoor (Administrators)
Follow-on access	Malware execution on the visitor's own machine	Hardcoded credentials granting passwordless admin access to the WordPress site itself, plus periodic fetches of additional injected JavaScript

The ClickFix technique — presenting a fake verification challenge that asks the visitor to run a system command — is one increasingly used in unrelated campaigns because it sidesteps browser and endpoint exploitation entirely, relying instead on the visitor's trust in what looks like a routine bot check [1][2]. The WordPress plugin path is arguably the more consequential of the two, because it converts a momentary CDN-level compromise into durable, site-specific persistence: even after Brevo revoked the stolen API key and removed the malicious Worker, any WordPress site whose administrator visited their dashboard during the exposure window could still be running the backdoored "Web Media Optimizer" plugin, with an authentication bypass baked directly into its code [1]. That plugin was fetched from `https://cdn10.sendibt1.com/p/wm.zip`, and Sansec's monitoring of a sample of twelve affected sites recorded 2,549 Content-Security-Policy violation reports during the active window, giving a rough sense of how aggressively the injected code attempted to load external resources [3].

Recommendations

Immediate Actions

Any organization that embeds Brevo's SDK, chat widget, or web forms should treat the September 14 window as an active-compromise period requiring direct verification rather than passive monitoring. Site operators should search their web server logs, CDN logs, and WordPress plugin directories for any of the published indicators — the `sendibt1.com` subdomain cluster, the `Web Media Optimizer` plugin name, references to `cdn10.sendibt1.com/p/wm.zip`, or the must-use plugins directory containing unfamiliar files — and should not assume that simply loading Brevo's current, clean JavaScript is sufficient evidence that no backdoor was installed during the exposure window [1][3]. WordPress site owners in particular should manually inspect the must-use plugins directory, since malicious code placed there does not appear in the standard plugin list and will not be caught by a routine admin-panel review.

Short-Term Mitigations

Security teams should audit every third-party script, tag manager, and CDN-embedded widget running on their own properties and confirm which vendors have direct edge-level or DNS-level write access to content served under their domain, since this incident demonstrates that a compromised vendor credential can bypass origin-side controls entirely. Where feasible, Subresource Integrity attributes should be applied to third-party script tags so that a browser refuses to execute a script whose content has changed unexpectedly, and Content-Security-Policy headers should be set and monitored independently of any CDN or edge-compute layer a vendor controls, so that a compromised Worker cannot silently strip them. Teams running WordPress sites that use Brevo integrations should rotate WordPress administrator credentials and review the admin user list for accounts created or modified during the exposure window.

Strategic Considerations

This incident is best understood as an instance of a broader and recurring pattern: a SaaS vendor's internal credential-management failure becoming a supply-chain event for every downstream customer that trusted its embedded code, without any action or misconfiguration on the customer's part. Organizations should extend vendor risk assessments beyond data-handling and compliance questionnaires to include concrete questions about how a vendor scopes and rotates infrastructure-provider credentials such as Cloudflare, AWS, or CDN API keys, since a single overprivileged, hardcoded key was sufficient to compromise every one of Brevo's customers simultaneously. Longer term, enterprises that rely heavily on embedded third-party JavaScript should evaluate architectural mitigations — sandboxed iframes, tag-management platforms with change approval workflows, or server-side tagging — that reduce the blast radius of a single vendor compromise from "every page view on our site" to a more contained failure domain.

CSA Resource Alignment

The Brevo incident sits squarely within a pattern CSA's AI Safety Initiative has been tracking across the SaaS ecosystem over the past year: a trusted, third-party vendor integration becomes the highest-leverage point of compromise precisely because it already carries legitimate, broadly scoped access into hundreds or thousands of downstream customer environments. CSA's research note "ShinyHunters' OAuth Pivot: A Year of SaaS Supply-Chain Breaches" examined this exact dynamic across the Salesloft/Drift, Gainsight, and Klue breaches, where attackers shifted from phishing individual employees to compromising a single vendor holding durable, dormant OAuth credentials across

hundreds of customer organizations at once [5]. The Brevo case is a close structural cousin: rather than an abused OAuth token, the vector was a hardcoded, overprivileged Cloudflare API key, but the underlying failure is the same — a long-lived, broadly scoped credential that a vendor's own engineering team lost track of, which an attacker could exploit to reach every downstream customer without needing to touch a single one of their environments directly. Both incidents suggest a common remediation gap: at least in these cases, credential lifecycle management for infrastructure-provider and platform-integration keys received less rigor than customer-facing authentication typically does.

CSA's AI Controls Matrix (AICM) v1.1 provides the applicable control baseline for both the vendor and the customer side of this relationship. The AICM's Identity and Access Management domain calls for scoped, time-limited credentials and regular access reviews — controls that, if applied to the Cloudflare API key implicated here, would likely have significantly limited how long a single leaked secret could carry full-account permissions. Its supply chain and third-party risk domains likewise call for downstream organizations to assess and monitor the credential-handling practices of vendors whose code executes directly in their production environments, which is precisely the gap this incident exposes for the tens of thousands of sites that had no visibility into how Brevo managed its own Cloudflare access. Organizations evaluating their exposure to this and similar incidents can use the AICM's control structure, available at cloudsecurityalliance.org/artifacts/ai-controls-matrix-v1-1, as a starting checklist for vendor security reviews going forward [6].

References

[1] Toulas, Bill. "Brevo supply-chain attack injected ClickFix scripts on customer sites." BleepingComputer, September 2026.

[2] Cyber Security News. "Brevo Supply Chain Attack Pushes WordPress Backdoors and ClickFix Malware to 100,000+ Sites." Cyber Security News, September 2026.

[3] Sansec. "Brevo supply chain attack hits 100k+ sites with WordPress backdoors and ClickFix malware." Sansec Research, September 16, 2026.

[4] SecurityWeek. "Brevo Supply Chain Attack Injects Malware Into 100,000 Websites." SecurityWeek, September 2026.

[5] Cloud Security Alliance AI Safety Initiative. "ShinyHunters' OAuth Pivot: A Year of SaaS Supply-Chain Breaches." Cloud Security Alliance, July 16, 2026.

[6] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance.