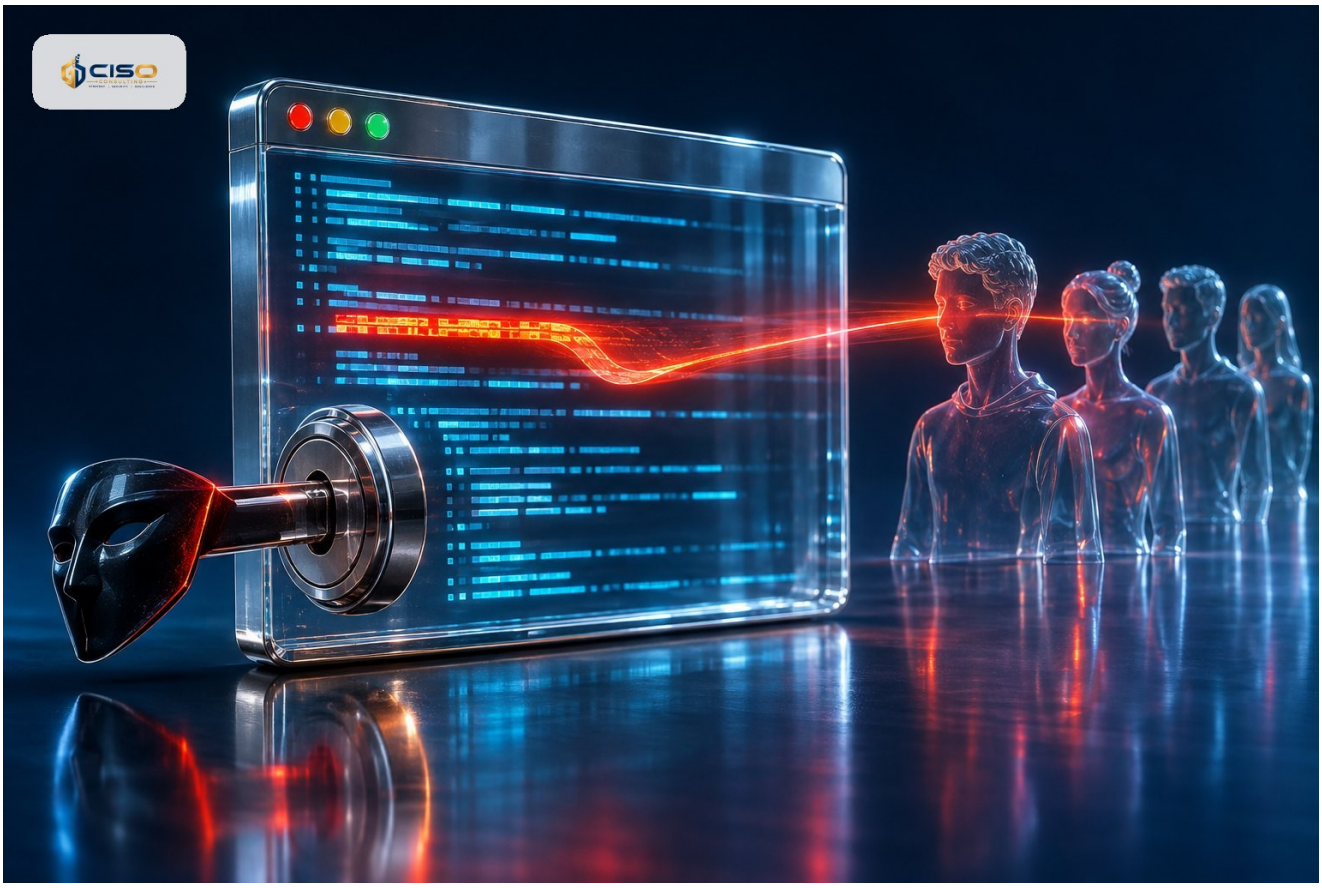
Optimizely Web Experimentation flaw (CVE-2026-132224) could let attackers alter site JavaScript



CISA lists CVE-2026-132224, a flaw in the account signup flow of Optimizely Web Experimentation that allows impersonation of customer users and changes to Project JavaScript, which then runs in visitors' browsers.

- The flaw is an external control of critical state data issue in the account signup flow.
- An attacker could impersonate existing customer users and modify Project JavaScript.
- Attacker-controlled code could be served through Optimizely-hosted assets and run in browsers of visitors to affected sites.
- CISA requires vendor mitigations, BOD 26-04 compliance and its forensics triage requirements; the federal due date is 2026-11-27.

> - If mitigations are unavailable, CISA directs discontinuing use of the product.

CISA has added CVE-2026-132224 to its catalogue. The issue affects Optimizely Web Experimentation and is classed as external control of critical state data in the account signup flow.

What the flaw allows

According to the source, an attacker could impersonate existing customer users and then modify Project JavaScript. Because that JavaScript is distributed through Optimizely-hosted assets, a successful attack could result in attacker-controlled code running in the browsers of people who visit the affected customer websites.

For organisations that use the product on public portals, this matters beyond the Optimizely account itself: the impact can reach customers and visitors of the website.

What CISA requires

CISA asks stakeholders to apply mitigations in line with vendor instructions and to meet its BOD 26-04 guidance on prioritising security updates based on risk, together with its forensics triage requirements. For cloud services, the applicable BOD 26-04 guidance should be followed, or use of the product should be discontinued if mitigations are not available. Stakeholders must assess each asset's internet exposure themselves. The federal due date is 2026-11-27.

The source does not describe exploitation in the wild, affected versions or specific fixes, so organisations should rely on the vendor's instructions for those details.

Suggested approach

Treat this as a web supply-chain risk: confirm whether the product is used, review who can change Project JavaScript, and check for unexpected changes.

What to do now

1. Identify all websites and portals that use Optimizely Web Experimentation, including those run by marketing or third parties.
2. Apply the vendor's mitigations and confirm status with the vendor in writing.
3. Review Project JavaScript for unexpected or unauthorised changes and check account and user activity for signs of impersonation.
4. Restrict and review which users can edit Project JavaScript, and enforce strong authentication on those accounts.
5. If mitigations are unavailable, plan to suspend use of the product and record the risk decision.
6. Preserve relevant logs and involve incident response if suspicious changes are found.