Solace Boats

Security case study — solaceboats.com

Sector	Boat manufacturer / WooCommerce
Platform	WordPress (Divi) + WooCommerce
Scope	File + Database
Attack family	Unauth-RCE fake plugin + JS injector
Entry vector	Compromised admin credentials (proven from activity log)
Verdict	CRITICAL Critical

> **Credential-theft compromise; RCE backdoor + JS injector; entry vector proven**

The problem

Solace Boats was serving “fake popups” to visitors and couldn't shake the infection even after installing security tooling. Underneath were two malicious files: a fake “page-ftp-sitemap” plugin that hid itself from the plugin list and exposed an unauthenticated remote-code-execution dispatcher with a magic-key admin backdoor, and a 116KB must-use plugin that injected heavily-obfuscated JavaScript to non-logged-in visitors only — the source of the popups.

What I did

- Completed a file-level forensic pass and decoded both malicious files — the AES-encrypted RCE dispatcher and magic-key backdoor in the fake plugin, and the visitor-only obfuscated JS injector in the must-use plugin.
- Analysed the database and, crucially, the WordPress Activity Log's 146,000 events to prove the entry vector: the fake plugin was installed under a legitimate but compromised admin account from an anomalous foreign IP — credential theft, not an exploit-install.
- Corrected the patient-zero date using the activity log rather than the (later, re-touched) file timestamp, and identified a second suspicious admin account and the attacker's ongoing control after security tooling was added.
- Documented the outstanding vulnerable components (file-manager, Ultimate Member, WooCommerce Payments, an exposed hardcoded API key) for rotation and patching.

The result

Unlike a database-only case, here I could prove rather than infer the entry vector: stolen admin credentials used to install a self-hiding backdoor that persisted even after Wordfence went in. That moved the recommendation decisively to server-level investigation, full credential rotation, and isolated hosting — and gave the client a defensible, timeline-backed account of exactly how their site was taken and why a surface clean would not hold.