

Heraldry Studies

Security case study — heraldrystudies.com

Sector	Specialist e-commerce (Shopify)
Platform	Shopify
Scope	Theme file audit + live-site investigation
Attack family	Fake-pixel data-harvester (custom pixel/app)
Entry vector	Shopify Custom Pixel or rogue app (not theme)
Verdict	CRITICAL Critical

Theme clean, but live site injected with a fake-pixel data-harvester

The problem

Heraldry Studies had two truths that looked contradictory: the theme export scanned completely clean, yet Google Ads had flagged the live site and named a suspicious domain. The live storefront was injecting a script site-wide under a fake “BIGO Pixel” comment — a 51KB obfuscated data-harvester that patched browser APIs to resist detection, harvested cookies, URLs, referrers and device fingerprints, and exfiltrated them to an external tracking endpoint.

What I did

- Scanned all 2,174 theme files first and confirmed them clean — then made clear a theme export can't see apps, pixels, or admin-level injections.
- Investigated the live site after the Google Ads flag and located the injected fake-pixel script in the storefront head, decoding its anti-detection, fingerprinting, and beacon-exfiltration behaviour.
- Proved it wasn't in the theme (absent from the export; the live theme unchanged for months) and identified the real source as a Shopify Custom Pixel or rogue app — so a theme reinstall would not fix it.
- Documented the access limitation honestly (the available login was a limited collaborator) and did not overstep into locked admin areas.

The result

The client received two reports that together solved the puzzle: a clean theme and a genuinely infected live site, with the malware correctly located outside the theme where a reinstall would never have reached it. I delivered concrete removal steps (Customer Events pixel → Apps →

orphaned script tags), account hardening, and the exact “made changes to comply” path to appeal the Google Ads flag — with an offer to re-verify once the owner could act from a full-access login.