Asian Studies Association of Australia

Security case study — asaa.asn.au

Sector	Academic association (AU)
Platform	WordPress + WooCommerce
Scope	Database + File
Attack family	Spam doorway kit + webshell + DB trigger
Entry vector	Nulled plugins
Verdict	CRITICAL Critical

> **Hands-on compromise; self-healing MySQL trigger found on re-scan**

The problem

This association's WooCommerce site had been worked over by hand: three rogue administrator accounts (one injected straight through SQL), a spam-cloaking doorway kit dropped into the uploads folder, a WP Console web-shell, and — worst of all — a copy of the site's wp-config credentials exfiltrated to a plaintext file anyone could reach. After the owner's team ran a cleanup, the attacker's admin account kept coming back.

What I did

- Mapped the full compromise across a 201MB database export and the file backup — the doorway kit, its command-and-control domains, the web-shell, and the exposed credential file.
- Traced the entry point to nulled (pirated) plugins bundled into the install.
- On the post-cleanup re-scan, kept digging after the rogue user rows were gone — and found the real reinfection engine: a malicious MySQL trigger on the comments table that silently re-created the rogue admin whenever a comment containing a specific magic phrase was posted.
- Gave the client the one-line fix to drop the trigger, plus the credential-rotation steps still outstanding.

The result

The cleanup had removed the visible malware but left the self-healing trigger in place — exactly the kind of thing row-deletion alone misses. Surfacing it turned a site that would have silently re-infected into one with a clear, final path to clean. It also became a permanent lesson in my playbook: on any admin-recreation case, always check the database for malicious triggers.