Mysterious Code.

AWS SELECT TIER SERVICES PARTNER
UK & EU · SINCE 2014

ANONYMISED ENGAGEMENT SUMMARY · AWS SECURITY RESPONSE

Security findings and remediation

An EU software company asked us to review the AWS environment behind its platform. Within two days, working with read-only access, the review became an incident response.

```
# a real engagement from autumn 2025. Details that could identify the client have been removed or generalised. Nothing has been invented, exaggerated or composited.
```

WHAT THIS IS

We publish this because **"what does a security engagement with you actually look like"** is a fair question, and this is the honest answer.

The client ran a **legacy application and a set of customer-facing e-commerce sites** on infrastructure the current team had inherited and did not fully understand. The ask was an architecture and DevOps review. What follows is what the review found, what we did about it, and how it ended.

WHAT THE REVIEW FOUND

Fifteen security findings. Six were critical.

1 Credentials served to anyone who asked
Configuration files were publicly accessible on live websites, exposing **AWS access keys and database, payment and email-service credentials** to anyone who requested them.

2 The production database open to the entire internet
Combined with the leaked credentials, anyone online could read, change or delete all of it. **We confirmed that ourselves, from outside.**

3 A leaked key already being abused
AWS had flagged malicious use of one exposed access key and had **suspended the account's email-sending service** as a result. This was not a theoretical exposure.

4 Backdoor malware in the application source
A backdoor file sat in the application code, **most likely planted through a file-upload feature**.

5 Authentication bypass
Any user account could be impersonated **by setting browser cookies**. No password needed.

6 Customer documents publicly downloadable
A directory of customer-uploaded documents was **open to anyone with the address**.

THE REMAINING NINE

- SSH open to the internet
- A backup user with full access to all stored files
- No monitoring or alerting of any kind
- Unencrypted volumes and snapshots
- An unmanaged, publicly reachable database replica
- Invalid SSL certificates across the customer sites
- Unused legacy code reachable in production
- No repeatable deployment process
- A single shared database behind every customer site

WHAT WE DID

DAY ONE

The day the client asked us to act, **the critical exposures were closed**:

- Security groups locked down, so **nothing but the load balancers faced the internet**
- Every exposed credential **rotated**, and the old ones disabled
- The malware **removed**, and two further suspicious files neutralised
- **AWS WAF** deployed in front of the sites, with logging enabled
- Public access to sensitive files **blocked server-wide**
- The exposed document directory **restricted**

WEEKS ONE AND TWO

Containment was completed:

- The stray, publicly reachable database replica **secured**
- Long-lived access keys replaced with **instance roles and least-privilege IAM**
- The team's own code remediation **reviewed** as it progressed
- A **target architecture** designed for a rebuilt platform: private subnets, managed databases, autoscaling and a proper CI pipeline, all defined as code

THE HONEST ENDING

Our recommendation was that the platform be **rebuilt, not patched**. Once malware has been present, no review can prove a second backdoor does not exist, and the application predated the team that now owns it.

The client chose instead to continue the remediation with their own team, working from our prioritised plan, with the critical exposures already closed and their changes reviewed with us as they went. That was their call to make, and we respect it. **We would rather hand over an honest plan than sell a rebuild.**

THE TAKEAWAY

Little of what we found was exotic. Exposed configuration files, open security groups, long-lived keys, missing monitoring: these are **the ordinary failure modes of infrastructure that outlives the people who built it**.

None of it was visible from inside the team, and all of it was findable in two days from read-only access. **That is what a second opinion is for.**

15	6	Day 1	2 weeks
SECURITY FINDINGS	CRITICAL	CRITICAL EXPOSURES CLOSED	CONTAINMENT COMPLETED

YOUR AWS

Wondering what a look at your AWS would find?

Start with a free second opinion, a 30-minute call with our founder. No account access needed, and you keep a short written read either way. Book a time or email us using the details below.

---

web mysteriouscode.com

mail hello@mysteriouscode.com

book savvycal.com/mysteriouscode/intro