BSides Birmingham 2026

From CSS to SysAdmin:
Anatomy of a CVSS 9.8 in National Education Infrastructure

BSides Birmingham 2026

> whoami

Systems Architect and Security Researcher
Identified a CVSS 9.8 in national infrastructure.

Rail Data Marketplace Hackathon 2026 Winner
Developed the *Sentinel* offline ticket validation engine.

Cisco Certified Cybersecurity Associate
Focusing on systemic logic flaws and infrastructural resilience.

> target

The Objective:

- An educational service relevant to millions of students in the UK
- Active in thousands of schools across the country
- Platform concerning Special Category PII

The Stack:

- **Frontend:** A support page
- **Middleware:** Transparent JSON-RPC Database Middleware
- **Backend:** Microsoft SQL Server

`[ API ]` → `[ JSON-RPC ]` → `[ MSSQL ]`

> recon

Discovery

- Client-available CSS metadata allowed internal identifiers to be available to low privileged accounts.
- CSS classes revealed the specific naming convention of the subjects on the platform.
- By identifying this pattern, I could enumerate every subject on the platform.

Logic

- Pattern: `[VENDOR]_[SUBJ]_[LVL]_[TYPE]`
- Example: `ID:ENG:3`

Impact

- This allowed me theoretical access to the entire content library.
- This led me to the realisation that the backend of this infrastructure was potentially as vulnerable as the frontend.

Methodology

If the platform was so unprotected, what happens when we talk directly to the interface?

That is how I discovered this vulnerability.

BSides Birmingham 2026

> gatekeeper

Architecture

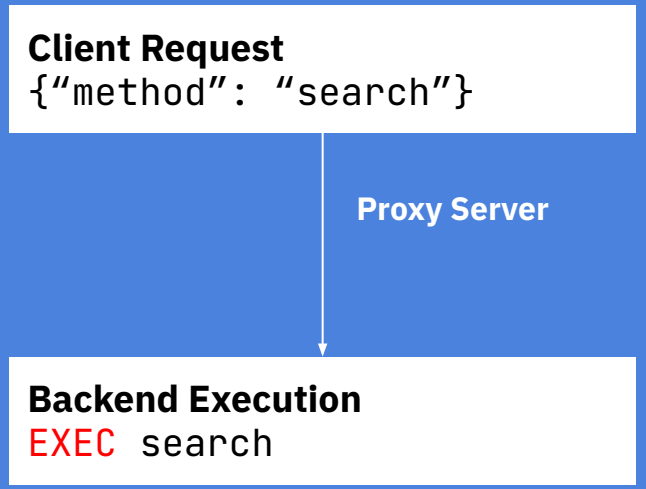
- A transparent database middleware designed to facilitate communication between the web frontend and the Microsoft SQL Server backend.

Defensive Measures

- Symbol blacklists appeared to be in-place in order to prevent simple SQL injection vulnerabilities.

Systemic Flaws

- Methods executed weren't checked against a whitelist.
- The proxy automatically prepended "EXEC " to any query, regardless of the method.
- This resulted in unauthorised access to system stored procedures.



> execution

Client Request
`{“method”: “sp_who”}`

Backend Execution
`EXEC sp_who`

Server Response

Information indicates the account running the query is the **database owner.**

> pivot

We can prove that we can run arbitrary system stored procedures.

But can we prove that we can use those procedures to execute arbitrary SQL statements?

By using the *method* and *params* arguments of the proxy, parameterised system stored procedures (such as `sp_sqlexec`) can be executed. This opens the door to more sophisticated attacks anchored towards SQL exploitation.

Client Request

```
{
  "method": "sp_sqlexec",
  "params": {
    "p1": "SELECT CAST(USER_NAME() AS INT)"
  }
}
```

Server Response

```
[
    “Conversion failed when converting the nvarchar value ‘Admin’ to data type int”
]
```

BSides Birmingham 2026

> blast_radius

The unauthenticated proxy backend provided access to a monolithic database structure.

Table enumeration via the `sp_tables` system stored procedure reveals that this database was used for important corporate and academic purposes.

All identifiable data has been anonymised.

Financial: `BankStatements`, `CreditNotes`, `Invoices`
Organisational: `users`, `usersSecurity`
Corporate: `HolidayRequests`, `DaysOff`
RCE Potential: `RemoteCommands`, `CommandGeneratorServers`
Academic: `[REDACTED]_Schools`

BSides Birmingham 2026

> disclosure

What happens when you find a vulnerability of this severity?

- Vendor had no Vulnerability Disclosure Policy.
- No public vulnerability report form - no security email.
- No triage team, instead redirected to helpdesk.
- Long email waiting times, opaque timelines.

But at least they fixed it right?

- Vendor dismissed this finding as “*low severity*”.
- *“Although you could gain access to one element of the system, your wider analysis was incorrect.”* - Vendor Team Lead

BSides Birmingham 2026

> escalation

If the vendor doesn't want to fix the issue, did I just leave it?

- Escalated to the **Government Cyber Coordination Centre (GC3)** via HackerOne to ensure legal and ethical compliance.
- I maintained a strict 90-day silent period to allow for silent remediation.
- Post-90-days, non-destructive testing confirmed the proxy remained fully unauthenticated and vulnerable.
- The **Government Cyber Coordination Centre (GC3)** overrode standard jurisdictional boundaries, granting an **individual exception** due to the severe supply-chain risk to the education sector, allowing for this severe hole in national education infrastructure to be dealt with appropriately.

BSides Birmingham 2026

> lessons_learnt

What lessons were learnt?

- Prepending execution commands to user input without a strict method whitelist is a systemic failure.
- SQL commands should never traverse across the client-server boundary.
- Failing to segment Corporate, Financial, and Academic databases means a single edge-case vulnerability results in total compromise.
- **In terms of disclosure:** When vendors become hostile, do not retaliate. Document the "machine truth," follow the 90-day protocol, and let the authority handle the rest.

BSides Birmingham 2026

> exit()

Mikey Whiston

mikey@mikeywhiston.dev