VECTRA™
An ENSIGN Company

VECTRA CYBER THREAT INTELLIGENCE

Your web apps are your front door. Here's what we keep finding.

Issue 2 | April 2026

DIRECTOR'S NOTE

Web applications are still the most reliably exploitable attack surface we see across Australian organisations. Over the past year our testing identified 324 unique findings across 1,809 individual instances. Close to a third of those were Medium severity issues in web apps and APIs. That's not a blip. That's a pattern.

This month we're looking at what keeps showing up in application security testing. Broken access controls, cross-site scripting, CSRF, and injection flaws. None of these are new. All of them still work. The Axios npm supply chain compromise in late March was a sharp reminder that your application is only as strong as its weakest dependency. A North Korean threat actor backdoored a library with 70 million weekly downloads through a single social engineering play. Every organisation should be asking hard questions about their third-party code.

The ASD and CISA have both called out internet-facing applications as a primary initial access vector for cybercriminals and state-sponsored actors. Our testing data backs that up. If your web apps haven't had an independent security assessment recently, the findings in this issue should give you a reason to book one in.

— **Shaun Burger** Director, Cyber Assurance | Vectra Corporation

THREAT SPOTLIGHT

Broken Access Controls: The Finding That Won't Go Away

Broken access controls topped the OWASP Top 10 in 2021 and stayed at number one in the 2025 refresh. That's not inertia. Access control logic is hard to get right. It's specific to each application and nearly impossible to validate with automated scanning alone. If you're relying on vulnerability scanners for your application security, you're missing the findings that matter most.

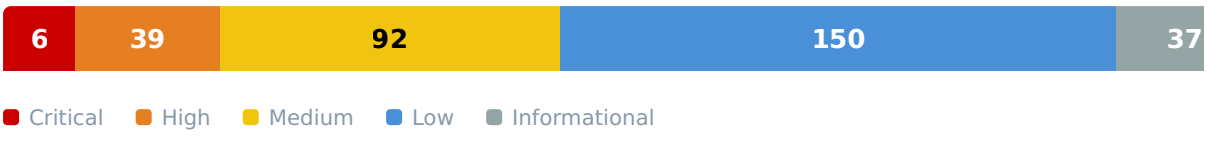
What does this look like in practice? An API endpoint that checks whether you're authenticated but doesn't check whether you're authorised to access the specific resource you're requesting. A user profile page where changing the user ID in the URL gives you access to someone else's data. An admin function that's hidden from the UI but still reachable if you know the endpoint.

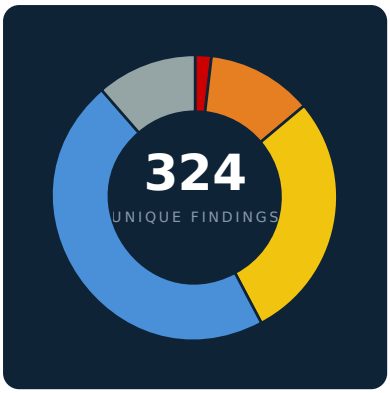
These aren't hypothetical. They're findings from the last three months across financial services, healthcare, and government clients. In one engagement, an IDOR vulnerability in a payments API let us view and modify transaction records for any customer in the system. The application had strong authentication, including MFA. But the authorisation model behind it was broken.

> **Key stat from our data: access control and authorisation flaws accounted for 17 findings across our 2025 engagements, with nearly half rated High or Critical.** These keep showing up across financial services, government, and professional services clients. CISA and the ASD jointly issued guidance on preventing web application access control abuse, calling IDOR vulnerabilities 'common, hard to prevent outside the development process, and exploitable at scale.' That's exactly what our testers find. The gap between authentication and authorisation is where most applications fail, and it takes manual, context-aware testing to catch it.

BY THE NUMBERS

324	17	42	9
UNIQUE FINDINGS	ACCESS CONTROL	AUTH / SESSION	XSS FINDINGS





Critical	6	1.9%
High	39	12.0%
Medium	92	28.4%
Low	150	46.3%
Informational	37	11.4%

TOP FINDINGS THIS PERIOD

Authentication / session management
42 instances · Medium

Information disclosure
28 instances · Low

Access control / authorisation (IDOR)
17 instances · High

Cross-site scripting (XSS)
9 instances · Medium

FROM THE FIELD

IDOR in payments API

Financial Services / Payments

A payment processing platform's API used predictable transaction IDs with no authorisation checks. Changing the transaction ID in API requests returned full details of other customers' payments, including card details and personal information. Authentication was strong. Authorisation at the API level was non-existent.

Stored XSS in customer portal

Professional Services

User-supplied input in a customer feedback form was rendered without sanitisation in the admin dashboard. An attacker could inject JavaScript that ran in the context of admin sessions, capturing session tokens or acting as an administrator. The fix was input validation and output encoding.

Verbose error messages leaking stack traces

Government / Transport

A government transport application returned full .NET stack traces and database connection strings in error responses. This information gave us the internal architecture, database technology, and version numbers to target specific known vulnerabilities. Disabling verbose errors in production took the development team 15 minutes.

Unsecured cloud database exposing 444,000 records

Financial Services / Fintech

In February 2026, Sydney-based fintech platform youX confirmed an unsecured MongoDB Atlas cluster had been open for roughly 10 months. 141GB of data was exfiltrated, covering 444,000 borrower records, 229,000 driver's licences, and 8,000 broker password hashes. A security researcher flagged the issue to youX in March 2025, but the exposure stayed open. This wasn't a sophisticated attack. It was an open database with no authentication, connected to over 90 downstream lenders. Cloud-hosted databases need the same access control rigour as any other part of your application stack. Regular configuration audits catch these before attackers do.

These findings are drawn from real Vectra engagements conducted between January 2025 and February 2026. Client identities have been anonymised.

HARDENING TIP

Lock down authorisation at the API layer, and audit your dependencies

Authentication tells you who someone is. Authorisation tells you what they're allowed to do. Most applications get authentication right and authorisation wrong.

The rule is simple: Every API endpoint must verify that the authenticated user is authorised to access the specific resource they're requesting. Not just 'is this user logged in' but 'does this user have permission to view/modify this specific record.' In practice, this means implementing object-level access controls. When a user requests /api/transactions/12345, your API must verify that the authenticated user owns or has permission to access transaction 12345. Don't rely on the UI to hide things. Assume every API endpoint will be called directly.

Test this by swapping authentication tokens between two different user accounts and attempting to access each other's resources. If it works, your authorisation model is broken. This is exactly the kind of testing our application security assessments are built around.

On the supply chain side: The Axios compromise showed that a single compromised dependency can backdoor your entire application. Run `npm audit` regularly. Pin your dependency versions in lockfiles. Review what your dependencies actually do, particularly postinstall scripts. Consider tools like Veracode to monitor for malicious packages. If you don't have visibility over what third-party code is running in your applications, a software composition analysis is a good place to start.

Not sure where to start? Our team can help you prioritise and validate these changes in your environment.

ON THE RADAR

Axios npm supply chain compromise (March 2026)

A North Korean threat actor (UNC1069) compromised the Axios npm package, which gets 70 million weekly downloads, by socially engineering the lead maintainer. Malicious versions deployed WAVESHAPER.V2, a cross-platform RAT, on every machine that installed the update. Google, Microsoft, and Palo Alto all published advisories. If your applications use JavaScript dependencies and you don't have a process for auditing your supply chain, this is the wake-up call. We can assess your software composition as part of a web application security engagement.

CISA and ASD joint advisory on IDOR vulnerabilities

CISA and the ASD's ACSC released joint guidance on preventing web application access control abuse, calling out insecure direct object references as 'common, hard to prevent outside the development process, and exploitable at scale.' We see the same thing in our testing. IDOR findings are some of our highest-impact results across financial services, healthcare, and government clients.

Critical web application CVEs: nginx-ui and n8n

Two critical vulnerabilities in widely used web management tools were actively exploited in March-April 2026. CVE-2026-33032 (CVSS 9.8) in nginx-ui allows authentication bypass. CVE-2026-27495 (CVSS 9.9) in the n8n workflow automation platform enables sandbox escape to remote code execution. If you're running internet-facing management interfaces, regular vulnerability assessments should be on the calendar.

OWASP Top 10:2025 released

The refreshed OWASP Top 10 keeps Broken Access Control at number one and moves Security Misconfiguration up to number two. A new category, Mishandling of Exceptional Conditions, comes in at number ten. The update shifts focus from symptoms to root causes, which is how we've always approached application security assessments.