SME Digital Advisor

Plain-English advice on digital risk

OWASP · 10 CARDS

OWASP Top 10 Flashcards

The OWASP Top 10 web application security risks, translated.

Reference site for SME owners. Free. No sign-up.

Companion to andrewreaassociates.com

smedigitaladvisor.co.uk

OWASP A01 · WEB APPLICATION RISK

Can the wrong person on your website see or change the wrong data?

The most common web-application flaw. About whether the website properly enforces who can do what once someone's logged in.

What you can do:
1. Ask your developer for the **access-control matrix**: who can see and do what.

2. Run the “different user” test: log in as a regular customer, try `/admin` or other customers' data.

3. For anything handling money or personal data, have an external pen test before launch and annually after.

OWASP A02 · WEB APPLICATION RISK

Is sensitive data scrambled properly when it's stored or sent?

Two things: data sent over the internet, and data stored on disk. Classic failures: HTTPS not enforced everywhere, passwords stored as plain text, sensitive data sat unencrypted in a database.

What you can do:
1. Test your site's TLS with SSL Labs. Aim for an A grade.

2. Ask your developer: “Show me how passwords are stored.” If they can decrypt them, the design is wrong.

3. Don't keep what you don't need. Delete old customer records on a schedule.

OWASP A03 · WEB APPLICATION RISK

Can someone trick your website into running their own commands?

The classic web flaw and still common: user-typed input passed directly to a database or system command.

What you can do:
1. Insist that parameterised queries (or the framework's ORM) are used everywhere.

2. Run a free automated scan with OWASP ZAP.

3. For business-critical applications, pay for a penetration test.

OWASP A04 · WEB APPLICATION RISK

Was the website designed to handle attack, or just good behaviour?

Some flaws can't be patched — they're built into the design. A password-reset flow with no rate limiting. A checkout that trusts the price the browser sends.

What you can do:
1. For new applications, ask: *“What did you think about misuse before features?”*

2. Run a five-minute “what would I try as an attacker?” exercise.

3. Add **rate limiting** on failed logins, password resets, account creation, contact forms.

OWASP A05 · WEB APPLICATION RISK

Are your servers and tools properly locked down?

Default passwords. Admin interfaces facing the internet. Verbose error messages. Cloud storage buckets accidentally public.

What you can do:
1. Run Security Headers on your site. Aim for B+.

2. Check admin paths (`/admin`, `/wp-admin`, `/phpmyadmin`). Restrict by IP or VPN.

3. For cloud users, use Defender for Cloud / AWS Trusted Advisor / Google SCC.

OWASP A06 · WEB APPLICATION RISK

Is your website built on software with known holes?

Most websites are built from many bits — CMS, plugins, libraries. If anything's out of date, the website inherits the holes.

What you can do:
1. For WordPress: use a **managed WP host** (WP Engine, Kinsta, SiteGround) that handles patching.

2. Ask your developer for an **SBOM**: which libraries and what version.

3. Set a monthly calendar reminder: review and apply CMS / plugin / library updates.

OWASP A07 · WEB APPLICATION RISK

Is your login system actually secure?

Classic mistakes: no rate limiting on failed logins, predictable password resets, sessions that never expire, no MFA.

What you can do:
1. Use a recognised authentication provider (Auth0, Microsoft Entra ID, AWS Cognito, Okta, Clerk).

2. Enforce **MFA** for any account handling money or personal data.

3. Set sensible password rules (long over complex), use modern hashing (bcrypt/argon2), rate-limit attempts.

OWASP A08 · WEB APPLICATION RISK

Can someone tamper with your software or data updates without you noticing?

How modern software updates itself. Auto-updates from untrusted sources or unverified library pulls can be poisoned.

What you can do:
1. Pin third-party library versions; don't auto-pull the latest.

2. Limit who can deploy code. Two-person review on production deploys.

3. If you write code, sign your releases. If you don't, ask vendors how they sign theirs.

OWASP A09 · WEB APPLICATION RISK

Would you know if your website was being attacked?

Average detection time is around 200 days. Usually because nobody was watching.

What you can do:
1. Log authentication, admin actions, and errors. Send logs off the server.

2. Set basic alerts: failed-login bursts, new admin users, bulk data exports.

3. For serious sites, a managed monitoring service should be watching out of hours.

OWASP A10 · WEB APPLICATION RISK

Can someone use your website as a proxy to reach private systems?

Some websites let users provide a URL. If the website blindly fetches that URL, an attacker can provide an internal URL and pull private data.

What you can do:
1. Block server-side requests to **internal IP ranges** (10.x, 172.16–31.x, 192.168.x, 169.254.x).

2. On AWS, ensure **IMDSv2** is required. Equivalents on Azure and GCP.

3. Where possible, **allowlist external destinations**.