

Risk-spotting cheat-sheet

Never face a blank register — questions and examples to surface what could go wrong.

USE IT DURING THE EXERCISE · RUN EACH ASSET THROUGH THE PROMPTS · DEPTH BEATS BREADTH

How to use this: take one asset at a time from your Assets tab. Run down the prompts below, and write anything that could plausibly go wrong into the register — one risk per row. Spend most of your time on your *most critical* assets. You won't use every prompt; you just need it to stop the blank page.

Three fast ways to find risks

- 1 · **Pre-mortem** — Imagine it's six months from now and this asset caused an incident. *What happened?* Work backwards from the disaster.
- 2 · **The C-I-A questions** — For each asset ask: what if its **Confidentiality**, **Integrity**, or **Availability** were lost?
- 3 · **STRIDE** (a security lens from Microsoft) — walk the asset through six threat types: **Spoofing** · **Tampering** · **Repudiation** · **Information disclosure** · **Denial of service** · **Elevation of privilege**.^[2]

Prompt catalogue — by risk category

Security

ASK Who can access this — and should they? Are secrets or keys exposed? Is input validated? Is data encrypted in transit and at rest? Is access logged?

EXAMPLES leaked database credentials · a user can see another user's data (broken access control) · injection · unencrypted PII · no audit trail. (See OWASP Top 10.^[1])

Technical / architectural

ASK Where's the single point of failure? What happens at 10× load? Is there a backup — and a *tested* restore? Are components too tightly coupled to change safely?

EXAMPLES one server with no failover · backups that were never test-restored · a shared library whose change breaks everything · a data model that can't evolve.

Operational / reliability

ASK How would you even know it broke? Who's on call? How do you deploy — and roll back? Do runbooks exist?

EXAMPLES no monitoring or alerting · manual, error-prone deploys · no rollback path · config drift between environments.

Third-party & supply chain

ASK Which vendors, APIs and open-source dependencies do you rely on? What if one is down, breached, or deprecated? Do you even know all your dependencies?

EXAMPLES a critical SaaS outage · a vulnerable open-source library (SOUP) · a compromised package in your build · a breaking third-party API change. (*Now a top OWASP 2025 category.*^[1])

Compliance / legal & data

ASK What personal data is involved, and what's the legal basis? How long is it kept? Where does it live — does it cross borders? Are your dependency licences compatible with your product?

EXAMPLES personal data with no consent or basis · indefinite retention · data leaving the EU · a copyleft licence in a proprietary product.

People / knowledge

ASK Who is the *only* person who understands this? Is it documented? What breaks if they leave, are ill, or on holiday?

EXAMPLES a bus factor of one · an undocumented system · no onboarding path · one overloaded person or team.

Schedule / delivery

ASK What must go right for the date to hold? What's the riskiest dependency? Where is the hidden or untested work?

EXAMPLES an optimistic estimate · blocked on another team · scope creep · no buffer left for testing or review.

AI-assisted development

ASK Do you review AI-generated code as if a junior wrote it? Could it be insecure or simply made-up? Are you pasting secrets or personal data into prompts? Who owns the licence / IP of generated code? What if the model or vendor changes or goes down?

EXAMPLES insecure generated code merged without review · prompt injection in an AI feature · secrets leaked into a prompt or log · over-reliance on one AI vendor · unclear IP on generated code.

THE SMELL TEST — IS IT WORTH LOGGING?

Write it down if it is (a) **plausible** in your context and (b) would **actually hurt**. Skip vague worries with no real impact. Five concrete, well-analysed risks beat twenty hand-wavy ones — and they're the ones you can actually score and mitigate.

Sources & where these come from

1. OWASP Top 10:2025 — *The ten most critical web application security risks* (incl. Broken Access Control, Security Misconfiguration, Software Supply Chain Failures). OWASP Foundation. owasp.org/Top10/2025/
2. Microsoft — *The STRIDE threat model* (Kohnfelder & Garg). Microsoft Learn. learn.microsoft.com/.../threat-modeling-tool-threats
3. NIST SP 800-30 Rev. 1 — *Guide for Conducting Risk Assessments* (threats, vulnerabilities, impact). NIST. csrc.nist.gov/pubs/sp/800/30/r1/final
4. ENISA — *Threat Landscape* (the EU agency's annual review of top cyber threats). enisa.europa.eu/topics/cyber-threats/threats-and-trends

A prompt list, not a full threat model — use it to get unstuck, then think about what's specific to your own system.