SECURE DEVHUB

FREE PRE-LAUNCH SECURITY MAP

# Who can break your app?

## Six kinds of attacker. Check what each one can reach before you launch.

Most checklists sort security by technology. This one sorts it by **who** could hurt you, so you can see the problem the way they would. 36 checks, each with a pass condition, a way to verify it, and its source.


| | | |
|---|---|---|
| 1 | The stranger · no account, just your URL | 7 checks |
| 2 | The customer · has a normal account | 7 checks |
| 3 | The insider · staff, admins, your own laptop | 6 checks |
| 4 | The supplier · packages, scripts, templates | 5 checks |
| 5 | The robot · bots, scripts and AI agents | 6 checks |
| 6 | The bad day · when something still goes wrong | 5 checks |

HOW TO USE THIS MAP

# Think like each attacker, one at a time

**Only check apps you own,** on your own accounts, preferably on a staging copy. Never test someone else's site.

## THREE SEVERITIES

| | |
|---|---|
| STOP | **Don't launch until it's fixed.** Someone could reach data, money or admin access with little effort. |
| FIX SOON | **Give it an owner and a date.** It needs an account, a trick, or bad luck to hurt you. |
| HARDENING | **Add it to the backlog.** An extra layer that makes the other mistakes less costly. |

## EVERY CHECK HAS THREE PARTS

- **What passing looks like**, in plain words.
- **How to check it**, usually from your own browser or dashboard.
- **Its source**: an OWASP or CWE ID you can look up for the detail.

## THE LAUNCH GATE

> **Ready to launch** = zero open **Stop** items, and every **Fix soon** item has an owner and a date. Re-check before each big release.

Companion guide: **Hack Your Own App First** walks through seven of these checks as two-minute attacks, with fix prompts for AI builders. Free at sanjayshah.com.np/security-checklists.

ATTACKER ONE

# 1 The stranger

No account, just your URL. Bots do this to every new site, all day.

- [ ] STOP **Your database isn't open to the internet.**
  Check: the database has no public endpoint; every Supabase/Firebase table has row-level security or rules.
  A01:2025 Broken Access Control

- [ ] STOP **No secret keys in anything the browser or app downloads.**
  Check: search your built JavaScript for key prefixes (Attack 1 in the companion guide).
  CWE-798 Use of Hard-coded Credentials

- [ ] STOP **Hidden files aren't served.**
  Check: `/.env`, `/.git/config` and `/backup.sql` on your own domain all return 404.
  A02:2025 Security Misconfiguration

- [ ] FIX SOON **Admin pages and API docs need a login.**
  Check: open `/admin`, `/swagger` and `/api-docs` in a private window.
  CWE-306 Missing Authentication for Critical Functionality

- [ ] FIX SOON **Errors don't explain your internals.**
  Check: break a URL on purpose; no stack trace, SQL or file path appears.
  CWE-209 · A10:2025 Mishandling of Exceptional Conditions

- [ ] HARDENING **Security headers are set:** HSTS, Content-Security-Policy, X-Content-Type-Options, frame-ancestors, Referrer-Policy.
  Check: run a header scanner against your own domain.
  A02:2025 Security Misconfiguration

- [ ] HARDENING **Redirect links only go to your own pages.**
  Check: `?next=https://example.com` keeps you on your site.
  CWE-601 URL Redirection to Untrusted Site

## 2 ATTACKER TWO

# The customer

Has a normal account, and a reason to see more, pay less, or get more than they should.

- [ ] **STOP** **Every record loaded by ID is checked against the signed-in user.**
  Check: as Alice, open Bob's order URL (Attack 2).
  **API1:2023 Broken Object Level Authorization · CWE-639**

- [ ] **STOP** **The server decides price, role, plan and "paid".**
  Check: edit the price in the checkout request and resend it (Attack 3).
  **CWE-602 Client-Side Enforcement of Server-Side Security**

- [ ] **STOP** **Orders unlock only after the payment provider's verified webhook.**
  Check: open the "payment success" URL without paying. Nothing unlocks.
  **A08:2025 Software or Data Integrity Failures**

- [ ] **FIX SOON** **Users can't add fields to their own update** (role, credit, team).
  Check: add `"role":"admin"` to a profile-update request.
  **CWE-915 Improperly Controlled Modification of Dynamically-Determined Object Attributes**

- [ ] **FIX SOON** **Plan limits are enforced by the server,** not just hidden buttons.
  Check: call a paid feature's API from a free account.
  **API5:2023 Broken Function Level Authorization**

- [ ] **FIX SOON** **Responses return only what the screen needs.**
  Check: read one API response in DevTools; no password hashes, internal notes or other users' emails.
  **API3:2023 Broken Object Property Level Authorization**

- [ ] **HARDENING** **Coupons, credits and trials can't be used twice** by sending the request twice at once.
  Check: fire the same redeem request twice in parallel.
  **API6:2023 Unrestricted Access to Sensitive Business Flows**

3

ATTACKER THREE

# The insider

Staff, admins, contractors, and your own laptop. Not always malicious: a stolen password works just as well.

- [ ] STOP **Every admin, cloud, payment, domain, email and code-hosting account uses MFA.**
  Check: the security page of each account.

- [ ] STOP **No demo admin or default password exists in production.**
  Check: try admin/admin-style logins on your live app.
  **CWE-798 Use of Hard-coded Credentials**

- [ ] FIX SOON **Production keys differ from test keys,** and live only in your host's secret settings.
  Check: compare the environment variables of each environment.
  **A04:2025 Cryptographic Failures**

- [ ] FIX SOON **Staff accounts have only the rights their job needs.**
  Check: list everyone who can export all customer data.
  **A01:2025 Broken Access Control**

- [ ] FIX SOON **Important actions are logged, without secrets:** logins, failed logins, role changes, exports, refunds.
  Check: do one, then find its log line.
  **A09:2025 Security Logging and Alerting Failures · CWE-532**

- [ ] HARDENING **Anyone who leaves loses access the same day.**
  Check: a written offboarding list exists and was used.

4

ATTACKER FOUR

# The supplier

Code you didn't write: packages, third-party scripts, templates and copied repos.

- [ ] STOP — **No dependency with a known critical or high vulnerability that has a fix.**
  Check: `npm audit`, `pip-audit` or OSV-Scanner.
  **A03:2025 Software Supply Chain Failures**

- [ ] FIX SOON — **Lockfiles are committed,** so every build installs the same versions.
  Check: `package-lock.json` or `pnpm-lock.yaml` is in git.
  **A03:2025 Software Supply Chain Failures**

- [ ] FIX SOON — **Third-party scripts are few,** from the official source, with Subresource Integrity where possible.
  Check: count the `<script src>` tags that load from other domains.
  **CWE-494 Download of Code Without Integrity Check**

- [ ] FIX SOON — **Templates and copied repos hide no instructions** for your AI assistant.
  Check: read AI rules files, README text and install scripts before your AI tool does.
  **LLM01:2025 Prompt Injection · LLM03:2025 Supply Chain**

- [ ] HARDENING — **Unmaintained packages are replaced.**
  Check: the last release date of each direct dependency.
  **CWE-1104 Use of Unmaintained Third Party Components**

## 5 ATTACKER FIVE

# The robot

Bots, scripts and AI agents. They don't get tired, and every call can cost you money.

- [ ] STOP **Login, sign-up, reset, OTP and every paid endpoint are rate-limited on the server.**
  Check: send the request 50 times in a loop (Attack 6).
  **API4:2023 Unrestricted Resource Consumption · CWE-307**

- [ ] STOP **AI features have per-user quotas and a hard spending cap at the provider.**
  Check: the provider's billing settings.
  **LLM10:2025 Unbounded Consumption**

- [ ] STOP **Content your AI reads can't give it orders,** and its tools only touch the current user's data.
  Check: hide an instruction in a document (Attack 7).
  **LLM01:2025 Prompt Injection · LLM06:2025 Excessive Agency**

- [ ] FIX SOON **Public forms (sign-up, contact, booking) have a bot check.**
  Check: submit the form from a script.

- [ ] FIX SOON **Model output is treated as untrusted:** escaped before display, never run as code or SQL.
  Check: ask your AI feature to reply with HTML or a script and see what the page does.
  **LLM05:2025 Improper Output Handling**

- [ ] HARDENING **Features that fetch a URL for the user block internal addresses.**
  Check: ask it to fetch `http://169.254.169.254/`.
  **API7:2023 Server Side Request Forgery · CWE-918**

## 6 ATTACKER SIX

# The bad day

Not a person: the moment something still goes wrong. What matters is how fast you recover.

- [ ] STOP **Backups run automatically, and you restored one in the last 90 days.**
  Check: the date of your last successful restore.

- [ ] FIX SOON **You can roll back to the previous version in minutes.**
  Check: time a rollback on staging.

- [ ] FIX SOON **Someone is alerted** on error spikes, login-failure spikes and downtime.
  Check: who received the last alert?
  **A09:2025 Security Logging and Alerting Failures**

- [ ] FIX SOON **Security researchers can reach you.**
  Check: `/.well-known/security.txt` exists with its two required fields, Contact and Expires.
  **RFC 9116**

- [ ] HARDENING **A short plan for telling affected users,** written before you need it.
  Check: who writes it, who approves it, how users are told.

QUESTIONS THE CODE CAN'T ANSWER

# Ask the owner

Any "no" or "not sure" becomes an action item.

1. Has any live key ever been pasted into a chat, ticket or AI tool? Rotate it.
2. Does every paid service (cloud, AI, SMS, email) have a budget alert?
3. Is the repository private, and was it ever public?
4. Who gets the alert at 2 a.m., and how?
5. Are you storing anything you don't need, such as card numbers, ID scans or plain passwords?
6. Which law covers your users' data where you sell (for example GDPR in the EU, or Nepal's Individual Privacy Act 2075), and does your privacy policy say what you collect and why?

# Launch gate

| ATTACKER | STOP OPEN | FIX SOON (OWNER + DATE) | HARDENING |
|---|---|---|---|
| 1 · The stranger | | | |
| 2 · The customer | | | |
| 3 · The insider | | | |
| 4 · The supplier | | | |
| 5 · The robot | | | |
| 6 · The bad day | | | |

**Ready** = zero open Stop items, and every Fix soon item has an owner and a date. **Honest limit:** this is a map, not a penetration test. If you take payments at scale or store health, ID or financial data, get a professional review too.

## SOURCES


- OWASP Top 10:2025, owasp.org/Top10/2025
- OWASP API Security Top 10 2023, owasp.org/API-Security
- OWASP Top 10 for LLM Applications 2025, genai.owasp.org/llm-top-10
- MITRE CWE definitions, cwe.mitre.org · RFC 9116 (security.txt), rfc-editor.org/rfc/rfc9116
- Go deeper: OWASP ASVS 5.0.0, asvs.owasp.org


**Watch the map**

"Who Can Break Your App?" on youtube.com/@SecureDevHub

**Try the attacks**

"Hack Your Own App First", free at sanjayshah.com.np/security-checklists