Gratis checklist + prompts

Veilig bouwen met AI

Bouwt u met Lovable, Bolt, v0, Cursor, Replit of Base44? AI bouwt snel, maar controleert niet of het veilig is. Deze twaalf punten loopt u zelf na, zonder technische kennis. Vink af wat klopt. Twijfelt u, vraag het dan aan uw bouwer of aan uw AI-tool.

Eerlijk vooraf
Deze checklist maakt de grootste risico's kleiner, maar is geen test. Of uw app echt dicht zit, ziet u pas als iemand het probeert. Achteraan leest u wanneer een test zinvol is.

1 Toegangsregels staan aan op elke tabel

Staan ze uit, dan kan iedereen met de publieke sleutel uit uw app alle gegevens lezen of wijzigen. Zo gaven in 2025 170 van 1.645 onderzochte Lovable-apps gegevens prijs.

- [ ] **Zo controleert u het:** Open in Supabase de Table Editor: elke tabel met klantgegevens toont "RLS enabled". Bij Firebase staat nergens "allow read, write: if true".

2 De regels kijken naar de ingelogde gebruiker

Een regel die altijd "ja" zegt, werkt alsof er geen regel is. De ingebouwde scan van een platform ziet dat verschil niet altijd.

- [ ] **Zo controleert u het:** Vraag uw AI-tool per regel uit te leggen wie wat mag. Staat er "using (true)" bij een tabel met klantgegevens, dan klopt het niet.

3 Met twee accounts getest

Alleen zo ziet u of klant A echt niet bij de gegevens van klant B kan.

- [ ] **Zo controleert u het:** Maak twee testaccounts. Log in als A en verander een nummer in de adresbalk naar iets van B. U hoort niets van B te zien.

4 Uitgelogd nagelopen

Een omgedraaide controle laat bezoekers zonder inlog juist alles zien. In een preview valt dat niet op.

- [ ] **Zo controleert u het:** Open uw app in een privévenster, zonder in te loggen, en klik overal doorheen. Ziet u gegevens die achter de inlog horen?

5 Geen geheime sleutels in de browser

Alles wat naar de browser gaat, kan iedereen lezen. Een geheime sleutel daarin geeft een vreemde dezelfde macht als u.

- [ ] **Zo controleert u het:** Open de paginabron of het tabblad Netwerk in uw browser en zoek op "service_role", "sk_live" en "secret". U hoort niets te vinden.

6 Betaalde API's alleen via de server

Een sleutel voor bijvoorbeeld een taalmodel of betalingen in de browser kan een ander gebruiken, op uw rekening.

- [ ] **Zo controleert u het:** Vraag uw bouwer of AI-tool waar elke sleutel staat. Het antwoord moet zijn: in een Edge Function of op de server, niet in de app.

7 Opslag is niet open voor iedereen

Bestanden zoals facturen of ID-bewijzen hebben eigen regels, los van de database. Een open opslag was de oorzaak van het lek bij de Tea-app.

- [ ] **Zo controleert u het:** Kopieer de link naar een geüpload bestand en open die in een privévenster. Laadt het bestand, terwijl het privé hoort te zijn? Dan staat de opslag open.

8 Test- en beheerpagina's zijn weg

AI-tools maken tijdens het bouwen handige snelkoppelingen die na de livegang blijven staan.

- [ ] **Zo controleert u het:** Probeer in een privévenster /admin, /debug, /test en /dashboard achter uw domein. U hoort een inlog of een nette foutpagina te zien.

9 Elk pakket bestaat echt

In een groot onderzoek bestond 19,7% van de pakketten die taalmodellen aanraadden niet. Aanvallers registreren zulke namen met schadelijke code erin.

- [ ] **Zo controleert u het:** Zoek elk nieuw pakket op npmjs.com of pypi.org op voordat het geïnstalleerd wordt. Onbekend of nauwelijks gebruikt? Niet installeren.

10 Testversies staan niet open op internet

Op sommige platforms staat een project na één klik op "publish" openbaar, ook als het nog een test is.

- [ ] **Zo controleert u het:** Controleer in de instellingen van uw platform of testprojecten op privé staan. Zet echte klantgegevens nooit in een testversie.

11 De AI-agent heeft beperkte rechten

Er zijn gevallen bekend waarin een AI-agent een productiedatabase wiste, ondanks de opdracht om niets aan te raken.

- [ ] **Zo controleert u het:** Laat de agent alleen op een testomgeving werken, houd test en live gescheiden en zorg voor een back-up die u kunt terugzetten.

12 Uw chatbot is herkenbaar en begrensd

Een chatbot is te sturen met slimme vragen (prompt injection) en kan op uw kosten onbeperkt draaien. Sinds 2 augustus 2026 moet een chatbot ook laten weten dat hij AI is.

- [ ] **Zo controleert u het:** Zet geen geheimen of klantgegevens in de systeemprompt, stel een limiet in bij uw AI-leverancier en meld bij het eerste bericht dat het om AI gaat.

Tien veilige prompts om te kopiëren

De prompts zijn in het Engels: AI-codetools volgen die het nauwkeurigst. Plak ze in Lovable, Bolt, Cursor of een andere tool. Nummer 1, 3, 4 en 5 komen letterlijk uit de documentatie van Lovable; de andere zijn gebaseerd op de genoemde bron.

1. Toegangsregels nalopen Na elke wijziging aan de database.

```
Review RLS policies to ensure users can only access their own data and shared data is
properly protected.
```

Bron: Lovable Docs, Security best practices

2. Toegangsregels goed opbouwen (Supabase) Als u nieuwe tabellen laat maken.

```
Check whether Row Level Security is enabled on every table in the public schema before writing any policy. For each table with user data, write separate policies for SELECT, INSERT, UPDATE and DELETE (no FOR ALL). Use auth.uid() to compare with the owner column, never current_user and never using (true). Explain each policy in one sentence.
```

Bron: Supabase Docs, AI-prompt voor RLS-policies

3. Zoeken naar gelekte sleutels Vóór elke livegang.

```
Review frontend code for exposed secrets, API keys, or sensitive information.
```

Bron: Lovable Docs, Security best practices

4. Een sleutel veilig toevoegen Als u een betaalde dienst koppelt.

```
Add an API key for Stripe payments securely without exposing it in the frontend code.
```

Bron: Lovable Docs, Security best practices

5. Controles naar de server Als uw app formulieren, prijzen of rechten controleert.

```
Move validation logic from React components to Edge Functions for better security.
```

Bron: Lovable Docs, Security best practices

6. Opslag dichtzetten Als gebruikers bestanden uploaden.

```
Check every storage bucket. Make buckets private unless the files are meant to be public, and add storage policies so users can only read and upload their own files, based on auth.uid().
```

Bron: Supabase Docs, Row Level Security

7. Betaalberichten controleren Als uw app webhooks krijgt van bijvoorbeeld Stripe.

```
Make sure every webhook handler verifies the signature with the provider's official library before processing the event, and rejects requests with a missing or invalid signature.
```

Bron: Stripe Docs, webhooks

8. Alleen bestaande pakketten Bij elk nieuw pakket.

```
Before adding any new package, confirm that it exists in the official npm or PyPI registry and name its official repository. Do not install a package you cannot verify.
```

Bron: USENIX Security 2025, package hallucinations

9. Opruimen vóór de livegang Vlak voordat u live gaat.

```
Before we go live: list and remove all debug, test and admin routes that are not protected by login, turn off source maps in production, and show a generic error page without technical details.
```

Bron: OWASP Web Security Testing Guide

10. De agent begrenzen Aan het begin van elke sessie met een AI-agent.

```
Work only on the development environment. Never run destructive database commands (DROP, DELETE, TRUNCATE) and never touch the production database. Ask me for explicit confirmation before any action that cannot be undone.
```

Bron: OWASP GenAI, LLM01 Prompt Injection

Eerlijk: een prompt vervangt geen test
Een goede prompt verkleint de kans op een fout, maar bewijst niets. De AI kan melden dat alles klaar is, terwijl er nog een regel staat die iedereen toelaat. De ingebouwde scan van Lovable keek in 2025 of een toegangsregel bestond, niet of hij werkte. Of uw app echt dicht zit, ziet u pas als iemand het probeert: met een tweede account, van buitenaf, zoals een aanvaller dat zou doen.

Wanneer laat u het testen?

De Scan

Website zonder inlog

Een scan die wij met de hand nalopen. Geen pentest. Vindt zichtbare sleutels, open beheerpagina's en onveilige instellingen.

De Inbraaktest (pentest website)

App met database of betalingen

Wij proberen met de hand van buitenaf bij uw gegevens te komen: publieke sleutel, open opslag, formulieren, betaalstappen.

De Inbraaktest Totaal (pentest plus)

Klanten loggen in, of er is een chatbot

Met testaccounts per rol: kan de ene gebruiker bij de gegevens van de andere? Ook API's, webhooks en uw AI-functie.

Altijd met toestemming
Wij testen alleen na schriftelijke toestemming van de eigenaar van het systeem. Draait uw app bij een platform zoals Lovable, Supabase of Vercel, dan testen we alleen uw eigen project, binnen de regels van dat platform.

Liever even overleggen?
Een intake van 30 minuten is gratis en vrijblijvend. Plan die via blackboxcyber.nl/diensten/ai-gebouwde-apps of bel +31 485 234 002. U krijgt na het gesprek één vaste prijs op papier.

Bronnen: Supabase Docs (Row Level Security; AI-prompt voor RLS-policies; API-sleutels) · Lovable Docs (Security best practices) · Superblocks, "Lovable vulnerability explained" (2025) · Spracklen e.a., USENIX Security 2025 · OWASP Top 10 for LLM Applications 2025 · Stripe Docs (webhooks). Links: blackboxcyber.nl/downloads/veilig-bouwen-met-ai
Versie 1.0 – oktober 2026. Deze checklist mag u vrij delen, ongewijzigd en met bronvermelding.