Sicherheitsrisiken von OpenClawd:

Prompt Injection

Wenn man dem Bot zu viele Rechte gibt, könnte er zum Beispiel eine E-Mail eines Angreifers lesen und automatisch Anweisungen ausführen, wie etwa vertrauliche Daten weitergeben oder Programme installieren.

Lösungen:

Klare Regeln, Inputs validieren, Bot nicht am eigenen System laufen lassen, Outputs und Aktionen des Bots nochmal bestätigen

Input-Validierung: Alle Eingaben werden vor der Verarbeitung gefiltert und validiert

Datenlecks und Datenschutz

Wenn man ClawdBot benutzt, können sensible Daten wie persönliche Infos oder Firmengeheimnisse auf fremde Server gelangen und dort eventuell abgefangen oder für das Training des Modells genutzt werden.

Lösungen:

- Daten verschlüsseln
- Keine sensiblen Infos eingeben
- Lokale Version für wichtige Daten nutzen
- Datenschutzrichtlinien prüfen

Schwache Zugangskontrolle

Wenn der Bot schlecht gesichert ist, können Unbefugte auf ihn zugreifen, z. B. Mitarbeiter ohne Rechte oder jemand mit geklautem Passwort.

Lösungen:

- Starke Passwörter & 2FA nutzen
- Nicht jeder Nutzer bekommt alle Rechte
- Zugriffe regelmäßig überprüfen

Gefährliche „Skills“ im Marketplace

Sicherheitsanalysen haben gezeigt, dass ein erheblicher Teil der OpenClaw-Skills im offiziellen ClawHub-Marktplatz kritische Fehler oder sogar Malware enthalten. Einige Skills können Passwörter, Tokens oder API-Schlüssel preisgeben.

Lösungen:

Recherchieren, bevor man „Skills“ nutzt

(auf Vertrauenswürdigkeit prüfen)

Gefahr beim lokalen Hosten

• Wenn der Bot zu viele Rechte hat, könnte er auf Dateien, Programme oder das ganze System einfach zugreifen

• Ein Angreifer könnte versuchen, den Bot auszunutzen und sensible Daten zu stehlen

Maßnahmen um OpenClawd „sicher zu machen“:

Das Gateway nur auf localhost (127.0.0.1) laufen lassen, nicht auf allen Netzwerkschnittstellen

Docker-Sandbox verwenden, sodass der Bot nur lesenden Zugriff auf seinen Arbeitsbereich hat

Authentifizierungs-Tokens und Pairing-Codes für alle Verbindungen verpflichtend machen

Risikoreiche Funktionen ausschalten, z. B. Shell-Befehle, Browsersteuerung oder Web-Abfragen

Externe Skills blockieren und nur vorher geprüften, manuell kontrollierten Code erlauben

API-Keys alle 90 Tage wechseln und in Umgebungsvariablen speichern, nicht in Konfigurationsdateien

Umfassendes Logging aktivieren und Echtzeitwarnungen bei verdächtigem Verhalten einrichten

Direktnachrichten nur im Pairing-Modus erlauben, keinen offenen Gruppenchat zulassen

Auf einer abgesonderten, dedizierten Maschine laufen lassen, ohne Zugriff auf produktive Systeme oder sensible Daten

Quellen

- https://owasp.org/www-project-top-10-for-large-language-model-applications/
- https://www.youtube.com/watch?v=BhjK2Gr0Ryc
- https://www.youtube.com/watch?v=1Y_u0fY-AbA
- https://www.aikido.dev/blog/why-trying-to-secure-openclaw-is-ridiculous
- https://www.theregister.com/2026/02/02/openclaw_security_issues/