KOSTENLOSER GUIDE • CLAUDE CODE

Sicherer Code – automatisch geprüft.

So installierst du das kostenlose **security-guidance** Plugin von Anthropic und lässt Claude deine Sicherheitslücken finden, während der Code entsteht – in unter 5 Minuten.

5 MIN SETUP €0 KOSTENLOS **ALLE** PLÄNE

! Bevor du startest

Drei Voraussetzungen, damit das Plugin vollständig läuft:

- **Claude Code CLI 2.1.144** oder neuer (prüfen mit `claude --version`)
- **Python 3.8+** im PATH (das Plugin nutzt `python3` für die Prüf-Engine)
- Du arbeitest in einem **Git-Repository** (für die Diff-Prüfung am Turn- und Commit-Ende)

> **Hinweis:** Beim ersten Lauf legt das Plugin eine kleine Python-Umgebung unter `~/.claude/security/` an. Dafür werden einmalig `pip` und Internet benötigt.

1 Plugin installieren

Öffne eine Claude-Code-Sitzung und gib ein:

```
/plugin install security-guidance@claude-plugins-official
```

Wähle bei der Abfrage **User-Scope** – dann lädt das Plugin in jeder neuen Sitzung auf diesem Rechner automatisch mit.

> Meldet Claude Code *„marketplace not found"*? Dann zuerst mit `/plugin marketplace add anthropics/claude-plugins-official` den offiziellen Marktplatz hinzufügen und erneut installieren.

2 Aktivieren

Ohne Neustart in der laufenden Sitzung aktivieren:

```
/reload-plugins
```

Fertig. Ab jetzt prüft das Plugin automatisch im Hintergrund – es gibt keinen Befehl, den du dir merken musst.

SO FUNKTIONIERT ES

3 Drei Prüf-Ebenen

Das Plugin schaut an drei Punkten auf deinen Code – jede Ebene tiefer als die vorige:

1 · Bei jeder Datei-Änderung

Blitzschneller Muster-Abgleich auf riskante Aufrufe (z. B. `eval(`, `.innerHTML =`, `os.system`). Kein Modell-Aufruf – kostet nichts.

2 · Am Ende jedes Turns

Eine separate Claude-Instanz prüft den gesamten Diff im Hintergrund – findet Dinge, die ein Text-Match nicht sieht: Autorisierungs-Lücken, Injection, SSRF, schwache Kryptografie.

3 · Bei jedem Commit / Push

Tiefe, agentische Prüfung, die umliegenden Code mitliest (Aufrufer, Sanitizer) – hält Fehlalarme niedrig und entscheidet, ob ein Fund echt ist.

Wichtig: Keine Ebene blockiert deine Writes oder Commits. Findet das Plugin etwas, bekommt Claude den Hinweis und behebt es direkt in der Sitzung. Es ist **eine** Verteidigungslinie – kein Ersatz für einen echten Security-Review.

WAS GEPRÜFT WIRD

SQL-Injection
Angreifer kommt übers Eingabefeld an deine Datenbank

Command-Injection
Eingaben steuern Befehle auf dem Server

XSS & DOM-Injection
Fremder Code läuft im Browser deiner Nutzer

Hardcoded Secrets
Passwörter & API-Keys offen im Quelltext

Unsafe Deserialization
z. B. `pickle` auf nicht vertrauten Daten

SSRF & Auth-Bypass
Server-seitige Requests & umgangene Rechte

Insgesamt rund 25 der häufigsten Schwachstellen-Klassen – nach OWASP-Logik.

4 Mehr rausholen

Im Team & in Cloud-Sitzungen aktivieren

User-Plugins gelten nur lokal. Für alle, die das Repo klonen (und für Claude Code im Web), trag es in die eingecheckten Projekt-Settings ein:

```
// .claude/settings.json
{
  "enabledPlugins": {
    "security-guidance@claude-plugins-official": true
  }
}
```

Eigene Regeln ergänzen

Lege `.claude/claude-security-guidance.md` an und beschreibe dein Bedrohungsmodell in Klartext – die Modell-Prüfungen lesen es als Zusatz-Kontext. Eigene Muster gehen in `.claude/security-patterns.yaml`.

Pausieren oder deaktivieren

- Ganz aus (ohne Deinstallation): Umgebungsvariable `SECURITY_GUIDANCE_DISABLE=1`
- Im User-Scope pausieren: `/plugin disable security-guidance@claude-plugins-official`
- Entfernen: `/plugin uninstall security-guidance@claude-plugins-official`

Kosten: Der Muster-Check ist gratis (kein Modell). Die Turn- und Commit-Reviews zählen wie normale Claude-Requests zu deinem Verbrauch. Verfügbar auf allen Plänen.