

# audit-context-building

# audit-context-building

“ Catalogue généré le 2026-05-11

## En une phrase

Avant de chercher des bugs, force Claude à lire le code ligne par ligne et à construire une compréhension profonde et structurée du projet — parce qu'on ne trouve pas de failles dans un code qu'on n'a pas vraiment compris.

## Quand l'utiliser

- Au démarrage d'un audit de sécurité sérieux, avant toute chasse aux vulnérabilités.
- Pour préparer une revue d'architecture ou un threat modeling (exercice qui imagine comment des attaquants pourraient s'en prendre au système).
- Quand tu veux éviter les conclusions hâtives basées sur « j'ai compris en gros ».
- Quand le code te paraît compliqué et que tu sens qu'une compréhension superficielle te ferait rater des choses.

## Comment l'invoquer

- **Slash command** : `/audit-context-building` (ou `/audit-context`).
- **Phrases déclencheurs (texte)** : "deep context", "audit prep", "build understanding before bug hunting", "line-by-line analysis".
- **Auto-invocation** : Sur demande explicite, au tout début d'une mission d'audit.

# Description détaillée

C'est un skill de discipline mentale, pas un outil. Il configure la manière dont Claude « pense » pendant la phase préparatoire d'un audit. Plutôt que de survoler le code à la recherche de patterns suspects (ce qui est tentant mais qui produit des hallucinations et des faux positifs), Claude bascule en mode « ultra-granulaire » : il lit fonction par fonction, bloc par bloc, parfois ligne par ligne.

Pour chaque morceau de code, il applique trois techniques. Le « First Principles » consiste à se demander ce que le code fait vraiment, sans accepter les apparences. Les « 5 Pourquoi » consistent à creuser une décision de design jusqu'à toucher la raison profonde. Les « 5 Comment » servent à explorer toutes les manières dont une logique peut être déclenchée ou contournée. À chaque étape, Claude note explicitement ses hypothèses, les invariants qu'il croit identifier, les acteurs en jeu (utilisateurs, admin, oracles), et il met à jour son modèle mental dès qu'une nouvelle preuve contredit ce qu'il pensait.

L'objectif n'est pas de trouver des bugs (ça vient après), mais de construire une carte mentale stable et explicite du système. Une fois cette carte établie, les autres skills d'audit (chasse aux vulnérabilités, analyse de variantes, contrôle de conformité à la spec) peuvent travailler sur des bases solides plutôt que sur des suppositions. Lent en apparence, mais beaucoup plus efficace au final.

## Pour aller plus loin

Pour les détails techniques, exemples et patterns spécifiques, voir le SKILL.md original.

## Source

- **Plugin** : `trailofbits/audit-context-building`
- **Nom interne** : `audit-context-building`
- **Fichier** : `/home/thymon/.claude/plugins/cache/trailofbits/audit-context-building/1.1.0/skills/audit-context-building/SKILL.md`

---

Revision #2

Created 2026-05-11 21:19:30 UTC by thymon

Updated 2026-05-11 21:37:08 UTC by thymon