audit-prep-assistant

audit-prep-assistant

> *Catalogue généré le 2026-05-11*

En une phrase

T'aide à préparer ton codebase pour un audit de sécurité externe en suivant la checklist Trail of Bits : objectifs clairs, analyse statique, nettoyage de code mort, tests, documentation — pour que les auditeurs travaillent vite et bien.

Quand l'utiliser

- 1 à 2 semaines **avant** un audit de sécurité externe d'un projet blockchain.
- Quand tu veux maximiser la valeur d'un audit (= ne pas payer des experts pour qu'ils découvrent des bugs triviaux que Slither aurait trouvé en 5 minutes).
- Pour donner aux auditeurs un point d'entrée propre : commit gelé, build qui marche, liste des fichiers in-scope.
- Quand tu veux écrire les "user stories" et flowcharts qui expliquent ce que ton protocole est censé faire (les auditeurs trouvent plus de bugs quand ils comprennent l'intention).
- Pour faire un dernier sweep de "low-hanging fruits" et de code mort avant de lancer le chrono d'audit.

Comment l'invoquer

- **Slash command** : `/audit-prep-assistant`
- **Phrases déclencheurs (texte)** : "prep for audit" / "get ready for security review" / "audit checklist"

- **Auto-invocation** : Sur demande explicite (généralement avant un audit planifié).

Description détaillée

Un audit de sécurité coûte cher (souvent 20-50k+ pour un protocole DeFi). Si les auditeurs passent 3 jours à comprendre comment builder ton projet ou à trier des findings Slither que tu aurais pu fixer toi-même, c'est de l'argent perdu. Trail of Bits a publié une checklist de préparation, et ce skill l'opérationnalise.

Le processus en plusieurs étapes : 1) **Set Review Goals** (le skill te pose des questions pour cadrer : quel niveau de sécurité vises-tu, quelles zones t'inquiètent le plus, quel scénario catastrophe veux-tu éviter), 2) **Resolve Easy Issues** (lance Slither pour Solidity, dylint pour Rust, golangci-lint pour Go, et t'aide à trier/fixer), 3) **Augmente la coverage de tests** et identifie le code non testé, 4) **Supprime le code mort** (fonctions/libs/features inutilisées), 5) **Rends le code accessible** (liste des fichiers in/out scope, instructions de build, branche dédiée, version freezée), 6) **Génère de la documentation** (flowcharts, user stories, commentaires inline).

C'est un skill **transversal** (pas spécifique à une chaîne) — il s'adapte à Solidity, Rust, Go, Cairo, etc.

Pour aller plus loin

Pour la checklist complète Trail of Bits et les commandes outil par outil, consulter le SKILL.md original à `/home/thymon/.claude/plugins/cache/trailofbits/building-secure-contracts/1.0.1/skills/audit-prep-assistant/SKILL.md`.

Source

- **Plugin** : `trailofbits/building-secure-contracts`
- **Nom interne** : `audit-prep-assistant`
- **Fichier** : `/home/thymon/.claude/plugins/cache/trailofbits/building-secure-contracts/1.0.1/skills/audit-prep-assistant/SKILL.md`

Revision #2
Created 2026-05-11 21:19:40 UTC by thymon
Updated 2026-05-11 21:37:17 UTC by thymon