secure-workflow-guide

secure-workflow-guide

> *Catalogue généré le 2026-05-11*

En une phrase

Pilote ton développement de smart contracts via un workflow sécurité Trail of Bits en 5 étapes (Slither, features spéciales, diagrammes, propriétés à fuzzer, revue manuelle) — à faire à chaque check-in ou avant déploiement.

Quand l'utiliser

- À **chaque check-in important** sur un projet de smart contracts (donc régulier, pas seulement avant l'audit).
- Avant un déploiement en mainnet.
- Quand tu veux une routine de sécurité qui couvre à la fois les outils auto (Slither, slither-prop, Echidna) et la revue manuelle des angles que les outils ratent.
- Pour structurer ta sécurité au-delà du simple "j'ai lancé Slither une fois".
- Quand le projet a des features spéciales (upgrades, tokens ERC, IBC...) et que tu veux les checks dédiés.

Comment l'invoquer

- **Slash command** : `/secure-workflow-guide`
- **Phrases déclencheurs (texte)** : "run secure development workflow" / "security check-in" / "pre-deployment review"
- **Auto-invocation** : Sur demande explicite (idéalement à chaque check-in).

Description détaillée

Trail of Bits a formalisé un workflow en 5 étapes pour maintenir un niveau de sécurité élevé tout au long du développement (et pas juste à la fin). Ce skill l'opérationnalise et l'adapte à ton projet (= il ne lance que ce qui s'applique).

Étape 1 — Check for Known Security Issues : exécute **Slither** (l'analyseur statique star de TOB, avec 70+ détecteurs intégrés pour Solidity), parse les findings par sévérité, trie les faux positifs avec toi. **Étape 2 — Check Special Features** : si tu as de l'upgradeabilité → `slither-check-upgradeability` (17 détecteurs de risques d'upgrade). Si c'est un token → `slither-check-erc` (conformité ERC) + redirection vers `token-integration-analyzer`. **Étape 3 — Visual Security Inspection** : génère 3 diagrammes (héritage, function summary, qui peut écrire dans quel storage) pour repérer visuellement les problèmes. **Étape 4 — Document Security Properties** : la phase la plus importante — t'aide à écrire les invariants (= "telle chose ne doit jamais arriver"), puis configure **Echidna** (fuzzer property-based) ou **Manticore** (exécution symbolique) pour les tester. **Étape 5 — Manual Review** : analyse les angles que les outils ne voient pas (privacy, front-running, MEV, cryptographie faible, hypothèses sur les oracles/flash loans).

C'est un workflow **récurrent** (à relancer souvent), contrairement à `audit-prep-assistant` qui est ponctuel.

Pour aller plus loin

Pour les commandes exactes de chaque étape et les explications détaillées, consulter le SKILL.md original à `/home/thymon/.claude/plugins/cache/trailofbits/building-secure-contracts/1.0.1/skills/secure-workflow-guide/SKILL.md` et la ressource `resources/WORKFLOW_STEPS.md`.

Source

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

Revision #2
Created 2026-05-11 21:19:36 UTC by thymon
Updated 2026-05-11 21:37:13 UTC by thymon