

# differential-review

# differential-review

“ Catalogue généré le 2026-05-11

## En une phrase

Compare deux versions d'un code (avant/après une modification) pour repérer ce qui a changé et identifier les risques de bugs ou de failles introduits par la modif.

## Quand l'utiliser

- Avant de fusionner une « pull request » (proposition de modification de code) sur ton dépôt.
- Quand un collègue ou Claude a touché à du code sensible (connexion, paiements, gestion de mots de passe).
- Pour vérifier qu'un « refactor » (réorganisation du code sans changer ses fonctionnalités) n'a pas cassé une protection au passage.
- Avant un déploiement en production sur un projet sérieux.
- Quand tu veux un rapport écrit qui justifie pourquoi tu valides (ou pas) un changement.

## Comment l'invoquer

- **Slash command** : `/differential-review`
- **Phrases déclencheurs (texte)** : "review this PR", "differential review", "check my diff", "pre-landing review".
- **Auto-invocation** : Sur demande explicite, ou suggérée automatiquement avant un merge.

# Description détaillée

Ce skill intervient au moment où une modification de code est sur le point d'être intégrée (un merge, un déploiement, une release). Il prend en entrée un « diff » — c'est-à-dire la liste des lignes ajoutées, supprimées ou modifiées — et le compare au reste du projet pour comprendre l'impact réel du changement.

Il ne se contente pas de lire les nouvelles lignes : il regarde aussi l'historique Git (qui a touché à ces fichiers avant, quand, pourquoi), calcule le « blast radius » (autrement dit : combien d'autres parties du code dépendent de ce que tu changes), vérifie si des tests automatiques couvrent les nouvelles lignes, et adapte sa profondeur d'analyse à la taille du projet (petit / moyen / gros).

En sortie, tu obtiens un rapport en Markdown qui liste les problèmes de sécurité potentiels, classés par niveau de risque (élevé / moyen / faible), avec pour chacun une preuve concrète (numéro de ligne, scénario d'attaque imaginé) et un niveau de confiance. Il sait notamment repérer les régressions de sécurité — par exemple, une vérification d'identité qu'on aurait silencieusement retirée en refactorisant.

## Pour aller plus loin

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

## Source

- **Plugin** : `trailofbits/differential-review`
- **Nom interne** : `differential-review`
- **Fichier** : `/home/thymon/.claude/plugins/cache/trailofbits/differential-review/1.0.0/skills/differential-review/SKILL.md`

---

Revision #2

Created 2026-05-11 21:19:29 UTC by thymon

Updated 2026-05-11 21:37:07 UTC by thymon