# cosmos-vulnerability-scanner

# cosmos-vulnerability-scanner

> *Catalogue généré le 2026-05-11*

## En une phrase

Scanne les modules d'une blockchain Cosmos SDK (en Go) ou les smart contracts CosmWasm (en Rust) pour repérer 9 types de failles critiques qui peuvent halter la chaîne, casser le consensus ou faire perdre des fonds.

## Quand l'utiliser

- Quand tu touches à une blockchain de l'écosystème Cosmos (Cosmos Hub, Osmosis, Injective, etc.) ou à un contrat CosmWasm.
- Avant de pousser un module custom (`x/monmodule/`) en prod.
- Pour vérifier qu'un contrat CosmWasm valide bien les "denoms" (= les noms de tokens — accepter un token bidon est une attaque classique).
- Quand tu intègres l'IBC (Inter-Blockchain Communication, le pont entre chaînes Cosmos) — des erreurs IBC peuvent vider des pools.
- Pour enquêter sur un crash de chaîne ("chain halt") après un upgrade.

# Comment l'invoquer

- **Slash command** : `/cosmos-vulnerability-scanner`
- **Phrases déclencheurs (texte)** : "audit my Cosmos module" / "check CosmWasm contract" / "scan IBC handler"
- **Auto-invocation** : Sur demande explicite (généralement lors d'un audit).

# Description détaillée

Cosmos est un écosystème de blockchains interopérables, où chaque chaîne est faite de modules Go (`x/banking`, `x/staking`...) ou héberge des contrats CosmWasm écrits en Rust. La grande particularité : si un module produit un résultat différent sur deux nœuds (= "non-déterminisme"), la chaîne s'arrête. Un simple `time.Now()` ou un map Go non trié peut tout faire planter.

Ce skill cherche 9 patterns : 1) **Missing Denom Validation** (accepter n'importe quel token), 2) **Insufficient Authorization** (oublier de vérifier qui envoie le message), 3) **Missing Balance Check**, 4) **Improper Reply Handling** (mal gérer les réponses de sous-messages), 5) **Missing Reply ID Check**, 6) **Improper IBC Packet Validation** (paquets cross-chain non validés — fuites de fonds), 7) **Unvalidated Execute Message**, 8) **Integer Overflow**, 9) **Reentrancy via Submessages** (modifier l'état avant un appel externe — réentrance possible).

Il cible aussi les méthodes ABCI (BeginBlocker, EndBlocker, CheckTx, DeliverTx) qui sont consensus-critiques, et utilise CodeQL si dispo pour traquer le non-déterminisme.

# Pour aller plus loin

Pour les détails techniques (exemples Go et Rust, patterns IBC, integration CodeQL), consulter le SKILL.md original à `/home/thymon/.claude/plugins/cache/trailofbits/building-secure-contracts/1.0.1/skills/cosmos-vulnerability-scanner/SKILL.md` et la ressource `resources/VULNERABILITY_PATTERNS.md`.

# Source

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

---

Revision #2
Created 2026-05-11 21:19:32 UTC by thymon
Updated 2026-05-11 21:37:09 UTC by thymon