libafl

libafl

> *Catalogue généré le 2026-05-11*

En une phrase

LibAFL est une "boîte à outils" en Rust pour construire ton propre fuzzer sur mesure quand les fuzzers tout faits (libFuzzer, AFL++) ne suffisent plus pour ton cas tordu.

Quand l'utiliser

- Construire un fuzzer personnalisé avec ses propres stratégies de mutation.
- Fuzzer une cible exotique (architecture rare, format de données très spécifique).
- Faire de la recherche académique en fuzzing (nouvelles techniques).
- Avoir le contrôle fin sur chaque composant du fuzzing (feedback, scheduler...).
- Remplacer libFuzzer en gardant le même code mais avec plus de puissance.

Comment l'invoquer

- **Slash command** : `/libafl`
- **Phrases déclencheurs (texte)** : "custom fuzzer", "advanced fuzzing", "LibAFL research"
- **Auto-invocation** : Sur demande explicite

Description détaillée

LibAFL n'est pas un fuzzer "clé en main" comme libFuzzer ou AFL++. C'est plutôt une bibliothèque Rust modulaire qui te donne tous les bouts d'un fuzzer (l'observer de couverture, le mutateur, le scheduler, le feedback…) et te laisse les assembler comme un Lego. Tu as donc une liberté quasi totale sur le comportement de ta machine à tester.

C'est l'outil pour les cas avancés. Par exemple : tu veux tester un firmware embarqué sur une puce inhabituelle, ou tu veux inventer une nouvelle stratégie de mutation qui n'existe nulle part ailleurs. Avec libFuzzer ou AFL++, tu serais coincé. Avec LibAFL, tu codes ce qu'il te faut. La contrepartie : la courbe d'apprentissage est raide (il faut connaître Rust et comprendre comment marche un fuzzer en interne).

Bon plan : LibAFL peut servir de "drop-in replacement" pour libFuzzer. Tu gardes ton harness existant, mais tu profites des features modernes (multi-cœur, mutations avancées) sans tout réécrire. C'est l'outil que les chercheurs en sécurité utilisent pour leurs papiers académiques et les pentesters pour leurs cibles les plus difficiles.

Pour aller plus loin

Pour les exemples concrets, options de configuration et patterns avancés, voir le SKILL.md original.

Source

- **Plugin** : `trailofbits/testing-handbook-skills`
- **Nom interne** : `libafl`
- **Fichier** : `/home/thymon/.claude/plugins/cache/trailofbits/testing-handbook-skills/1.0.1/skills/libafl/SKILL.md`

Revision #2
Created 2026-05-11 21:19:55 UTC by thymon
Updated 2026-05-11 21:37:29 UTC by thymon