fuzzing-dictionary

fuzzing-dictionary

> *Catalogue généré le 2026-05-11*

En une phrase

Un dictionnaire de fuzzing, c'est une liste de "mots magiques" qu'on donne au fuzzer pour qu'il sache de quoi il parle (mots-clés HTTP, balises XML, octets de header PNG…) et trouve des bugs plus vite.

Quand l'utiliser

- Fuzzer un parseur de format binaire (PNG, ZIP, PDF) qui a des "magic bytes" en en-tête.
- Fuzzer un parseur de format texte (JSON, XML, YAML, configs).
- Fuzzer un protocole réseau (HTTP, DNS, protocole maison).
- Quand ton fuzzer plafonne sans trouver de nouveaux chemins (il bloque sur une validation).
- Pour rendre n'importe quelle campagne de fuzzing plus efficace (format-aware fuzzing).

Comment l'invoquer

- **Slash command** : `/fuzzing-dictionary`
- **Phrases déclencheurs (texte)** : "fuzzing dictionary", "format-aware fuzzing", "magic bytes"
- **Auto-invocation** : Sur demande explicite

Description détaillée

Quand un fuzzer génère des entrées au hasard, il a très peu de chances de produire spontanément la string `"Content-Type"` ou les 4 octets magiques `89 50 4E 47` qui ouvrent un fichier PNG. Du coup, il reste bloqué dans les premières lignes du parseur ("le format est invalide, retour") sans jamais explorer le cœur du code.

Un dictionnaire de fuzzing résout ce problème. C'est un simple fichier texte avec des "tokens" entre guillemets, par exemple `"GET "`, `"Content-Length:"`, `kw="\xFF\xD8\xFF\xE0"` (entête JPEG). Tu le donnes au fuzzer avec l'option `-dict=`, et il va insérer ces morceaux dans ses mutations. D'un coup, ses entrées ressemblent vaguement à du vrai protocole/format, et il franchit la couche de validation pour aller explorer en profondeur.

Le format est universel : un même dictionnaire marche avec libFuzzer, AFL++ et cargo-fuzz. Tu peux trouver des dictionnaires prêts à l'emploi sur le repo AFL++ (HTTP, JS, XML, etc.) ou en construire un toi-même en extrayant les chaînes de la doc du format. C'est une des optimisations à plus fort retour sur investissement pour booster une campagne de fuzzing — souvent x10 ou x100 sur la vitesse de découverte de bugs.

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** : `fuzzing-dictionary`
- **Fichier** : `/home/thymon/.claude/plugins/cache/trailofbits/testing-handbook-skills/1.0.1/skills/fuzzing-dictionary/SKILL.md`

Revision #2
Created 2026-05-11 21:19:46 UTC by thymon
Updated 2026-05-11 21:37:21 UTC by thymon