Fuzzing as a service (FAAS)

Frédéric Vachon, Martin Grogan et Benjamin Rosa

Plan de la présentation

- Fuzzing
- État actuel du Fuzzing
- Justification et nature du changement
- Présentation de Fuzzing as a service (FAAS)

Fuzzing

- Technique pour tester des logiciels basé qui l'injection de données aléatoires dans un programme
- Différents types de fuzzing :
  - Applications web
  - Protocoles réseaux
  - Fichiers exécutables
- Cette présentation sera axé sur les Fuzzers de fichiers exécutables

État actuel du fuzzing

PeachFuzzer

- Version communautaire et version entreprise payante
- Supporte :
  - Drivers
  - Protocoles réseaux
  - Systeme embarqué
  - ...
- Interface web (professionel)
- Peach Pits
  - Tests préexistants qui permet de tester des failles communes

American Fuzzy Lop

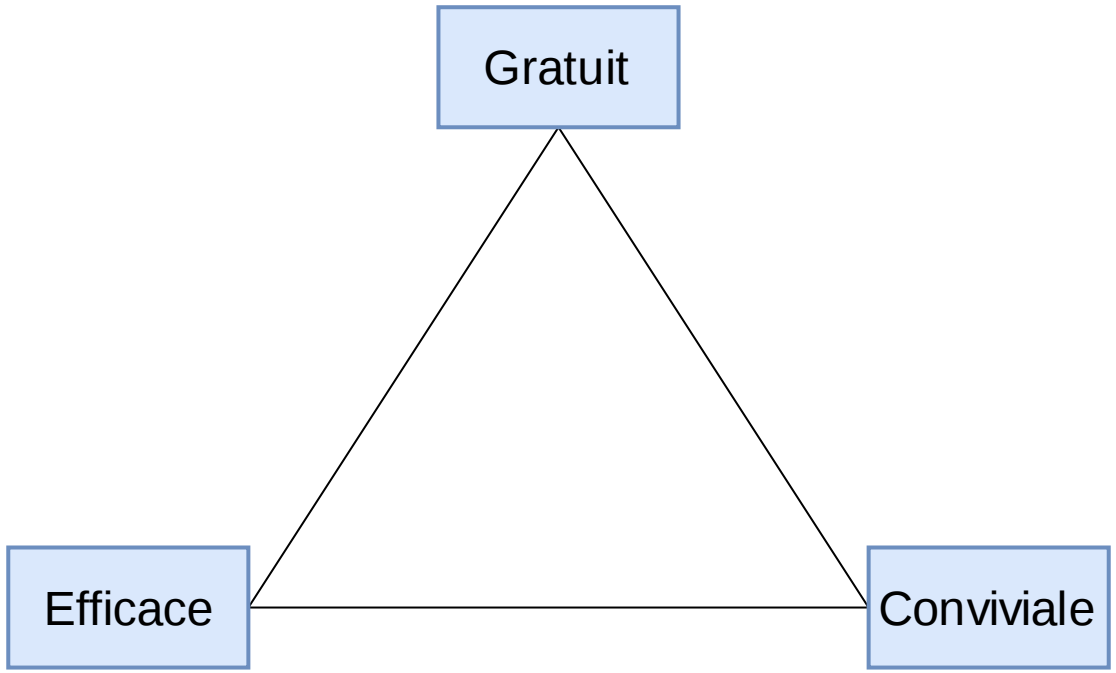
- Orienté pour la recherche de failles de sécurité
- Demande peu de configuration
- Interface peu conviviale en ligne de commande
- Pas d'interface web

État actuel des fuzzers

- Deux catégories
- Catégorie 1
  - Gratuité
  - Interface difficile d'utilisation
- Catégorie 2
  - Produit payant
  - Interface conviviale

Justification et nature du changement

Triangle de l'offre actuel



Nature du changement

- Nous voulons créer un produit qui pourra répondre à ces 3 critères
- Offrir une interface web conviviale
- Avoir un produit complètement gratuit
- Logiciel libre
- Notre logiciel sera efficace en terme découverte de bugs
- Nous supporterons les programmes compilés pour la plateforme Linux

Autres changements

- Environnement opérationnel
  - Démocratisation de l'utilisation du fuzzing
  - Nouveaux utilisateurs à former
  - Solution : Interface web conviviale
- Support
  - Le support sera assuré par la communauté
  - Un forum de discussion sera créé pour répondre à ce besoin

Changements essentiels

- Fuzzer des applications prenant des entrées en ligne de commande
- Interface web facile d'utilisation
- Pouvoir rédiger les fichiers en entrée (hex editor)
- Consulter les tâches de fuzzing (en cours, à l'arrêt, terminée)
- Consulter les rapports de crash
- Lancer plusieurs activités de Fuzzing simultanément

Changement essentiels (suite)

- Utiliser la technologie des conteneurs de Docker plutôt qu'une machine virtuelle
- Avoir un répartiteur qui va répartir les tâches de Fuzzing entre les conteneurs
- Base de données qui va entreposer les rapports de crash
- Pouvoir chercher un rapport de crash dans la liste des rapports
- Pouvoir chercher une tâche de fuzzing dans la liste des tâches

Présentation de Fuzzing as a service (FAAS)

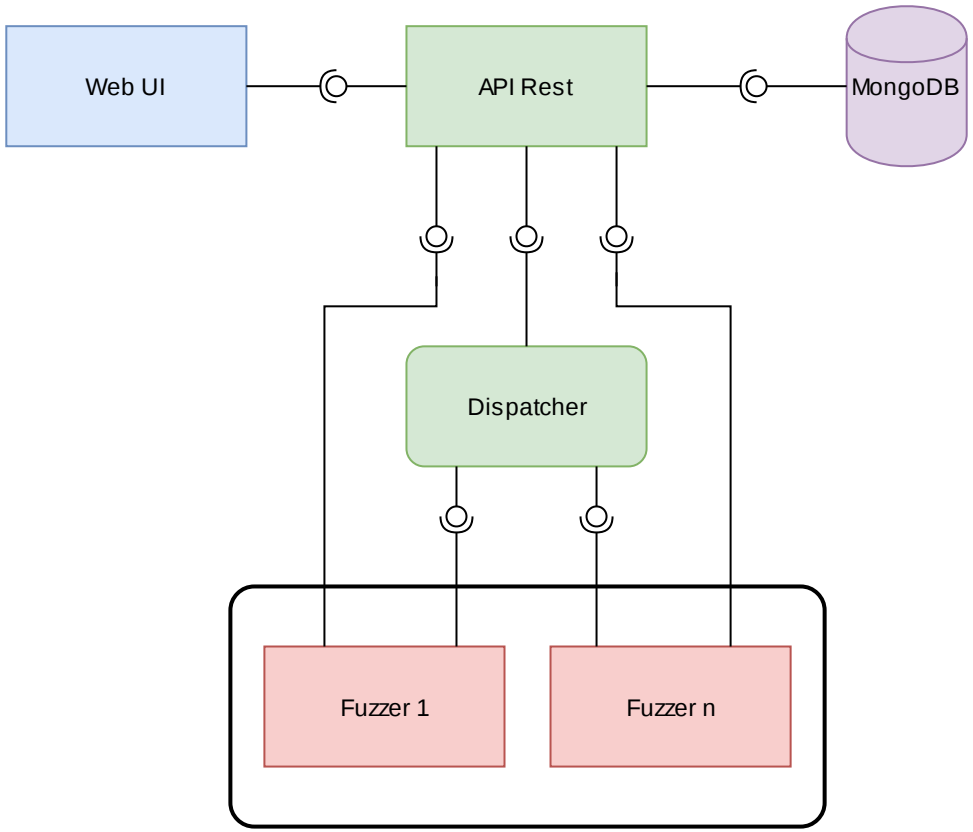
Objectifs

- Vise les développeurs indépendants et les petites à moyennes entreprises
- Améliorer la qualité et la sécurité des logiciels pour les acteurs qui n'ont pas d'immense budget à investir
- Inciter les gens à intégrer le fuzzing dans le cycle du développement de leur logiciel
- Démocratiser la pratique du fuzzing
  - Produit gratuit
  - Facile d'utilisation

Politiques opérationnelles et contraintes

- Deux contraintes importantes :
  - Architectures supportées :
    - Support des architectures x86 et x86_64 uniquement
    - Pas de support pour MIPS, ARM, PowerPC, etc.
  - Plateformes supportées :
    - Support de la plateforme Linux uniquement
    - Pas de support pour Android, MacOS, iOS, Windows, etc.

Principales composantes du système



Fonctionnalités de FAAS

- Fuzzer les données en entrée standards
- Fuzzer les arguments du programme
- Fuzzer les fichiers lus par le programme
- Fuzzer les variables d'environnement
- Créer, éditer, supprimer une ou plusieurs tâche de fuzzing depuis l'interface web
- Consulter la liste des rapports de crash depuis l'interface web
- Consulter la liste des tâches de fuzzing (celle à l'arrêt, en cours ou terminée).

Fonctionnalités de FAAS (suite)

- Se connecter et se déconnecter sur son compte utilisateur
- Stocker les rapports de crash dans une base de donnée
- Lancer plusieurs activités de fuzzing au moyen d'un conteneur Docker
- Créer ses propres données d'entrée au moyen d'un Hex Editor, via l'interface web
- L'utilisateur doit pouvoir consulter et mettre à jour ses données de profil
- Pouvoir chercher une tâche de fuzzing dans la liste des tâches
- Pouvoir chercher un rapport de crash dans la liste des rapports

Conclusion

- FAAS
  - Démocratiser le Fuzzing
  - Intégrer le Fuzzing dans le cycle de vie du logiciel
  - Solution conviviale, efficace et gratuite