HEIG-VD HAUTE ÉCOLE D'INGÉNIERIE ET DE GESTION DU CANTON DE VAUD

Prof. Yann Thoma

Exercices du cours VSE
Exercices de vérification logicielle
Fuzzing
semestre automne 2024 - 2025

Cet exercice vise à explorer deux frameworks pour le fuzzing *automatique*, cifuzz et clang, ainsi que ce qui peut être mis en place en python pour du fuzzing de fichiers XML.

cifuzz

Reprenez le code du dossier cifuzz, et dans un terminal placez-vous dans ce répertoire.
Lancez la commande :

```
cifuzz run my_fuzz_test
```

Cette commande compile les sources et lance un test de fuzzing.
Il est également possible d'exploiter le `CMakeLists.txt` présent dans le répertoire :

```
mkdir build
cd build
cmake ..
make
./cifuzz_example        # Lance le programme correspondant a main.cpp
./my_fuzz_test          # Lance le programme de fuzzing, mais sans fuzzing
cifuzz run my_fuzz_test # Lance les tests avec fuzzing
```

Vous pouvez modifier le fichier `exploreme.cpp` pour ajouter/enlever des bugs.

clang

`clang` vient avec un fuzzer. Vous disposez d'un exemple dans le répertoire `clangfuzzing`. Le script `compile.sh` permet de compiler le fichier source. Vous pouvez y trouver à l'intérieur `-DTEST1` qui, si vous le modifiez, permet de tester différentes erreurs (regardez le fichier source pour voir de quoi il s'agit). Lancez à chaque fois l'exécutable. Pour `TEST3` vous devriez observer quelque chose de surprenant.
Faites quelques essais de modification du code pour voir ce qui est détecté ou non.

XML fuzzing

Dans le cadre d'un projet de pharmacologie (`https://www.tucuxi.ch`), nous avons développé un wrapper python autour du coeur de calcul des prédictions de concentration de médicament dans le sang. Le coeur de calcul peut prendre en entrée un fichier XML avec des données sur le patient, son traitement, des mesures de concentrations et des données cliniques, ainsi que le type de calcul à effectuer. Evidemment nous aimerions être sûrs que le code ne plante pas lamentablement si le XML est mal formé ou contient de données hors des spécifications.
Pour ce faire rien de tel que du fuzzing de fichier XML. Dans le répertoire xmlfuzzing vous trouvez un script python, `tucuclifuzzing.py` qui charge le fichier `imatinib.tqf`, et lance un calcul avec à chaque fois une modification du fichier original. Pour ce faire il parcours l'arbre XML et y remplace les entiers par leur valeur négative.
Avant de pouvoir faire fonctionner le script il faut une ou deux étapes d'installation. Ci-dessous la liste des commandes à faire, dans la VM (je vous laisse adapter pour votre machine le cas échéant), depuis le répertoire xmlfuzzing (qui ne doit pas être un dossier partagé avec le host) :

```
sudo apt install python3.10-venv
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

A chaque fois que vous voudriez relancer un terminal pour des tests, il faudra exécuter

```
source venv/bin/activate
```

En lançant le script vous devriez observer l'exécution de plusieurs calculs. Vous devriez également voir la création d'un répertoire `output` et son peuplement par tous les fichiers générés à partir de l'original.
Nous vous demandons d'ajouter deux ou trois de ces modifications :

1. Faire que les champs contenant des nombres à virgule soient remplacés par la même valeur mais négative.
2. Faire que les champs numériques (entier ou flottant) soient remplacés par 0.
3. Faire que les champs numériques (entier ou flottant) soient remplacés par 1000 fois leur valeur.
4. Faire que les champs *unit* soient modifiés avec une chaine aléatoire quelconque (qui n'est pas une unité).
5. Faire que les champs *unit*, s'ils sont un poids, deviennent une concentration, et réciproquement :
   — kg $\rightarrow$ kg/l
   — mg $\rightarrow$ mg/l
   — mg/l $\rightarrow$ mg
6. Faire que les dates soient décalées dans le temps de plus de 2 mois
7. Faire que les dates soient décalées dans le temps de moins de 2 moins

Saurez-vous faire planter le système ?