

Mémo

Jonathan DUBOIS

March 25, 2018

1 Tests

1.1 Couverture des tests

Afin de mesurer la couverture de l'exécution du programme, nous utilisons `lcov` et `genhtml`, `lcov` sert à montrer quelles lignes du code ont été parcourues lors des exécutions du programme depuis la compilation, `genhtml` sert quand à lui à générer un répertoire html permettant de visualiser graphiquement le rapport effectué par `lcov`.

Commandes :

- Options de compilation :

```
CFLAGS = -fprofile-arcs -ftest-coverage
```

- Génération du rapport `rap.info` :

```
lcov --base-directory ./src/ --directory ./src -c -o rap.info
```

- Génération du rapport en html :

```
genhtml rap.info --output-directory out
```

1.2 Génération des tests

2 Détection de défaut et erreurs

2.1 Klee

Afin d'utiliser klee, il faut dans un premier temps modifier légèrement le programme, en remplaçant les entrées (clavier, fichier, rand) par `klee_make_symbolic()`

Ensuite, il faut compiler son projet avec Clang:

- Génération des dépendances :

```
clang -Wall -Werror -DKLEE -emit-llvm -g -c FICHIER.c
```

- Génération du programme :

```
llvm-link DEPENDANCES.bc -o appli.bc
```

On lance ensuite Klee sur appli.bc :

```
klee appli.bc
```

2.2 Valgrind

L'outil Valgrind sert principalement à analyser la mémoire, cependant il peut aussi repérer certaines erreurs comme Klee. Pour l'utiliser il suffit de lancer les commandes suivantes, qui vont créer un fichier log.txt dans lequel seront notés les différentes erreurs trouvées :

3 Analyse des performances

3.1 Gprof

Gprof sert à donner des informations sur le temps d'exécution du programme. Pour l'utiliser il faut compiler avec l'option -pg dans les CFLAGS et les LDFLAGS. L'exécution du programme générera un compte rendu gmon.out, pour le consulter il suffira de lancer la commande :

```
gprof appli
```

3.2 perf

Perf nous permet de réaliser deux analyses, les problèmes de cache, et l'occupation du processeur en temps réel. Erreur de cache :

```
sudo perf stat -e cache-misses,cache-references ./appli
```

Occupation processeur :

- Lancer le programme en tâche de fond
- Récupérer le PID
- Dans un autre terminal, lancer `perf top -p PID`
- Passer le programme en premier plan

4 Vulnérabilités