Fuzzing

Выполняется одно из заданий на выбор. В случае выполнения второго задания: к финальному результату теста в конце курса прибавляется 1 балл по 5-балльной шкале (возможно, после устной беседы по некоторым особенностям реализации второго задания).

Уронить приложение

В каталоге afl-instrumented выложен исполняемый файл формата ELF, собранный под архитектуру AMD64 без дополнительных библиотек-зависимостей, кроме системных, типа libc.6. Программа принимает на вход 1 параметр - строку из стандартного потока ввода, и считывает из файла `generated-seed` число:

```
./buggy-soft
"test string"
```

Файл инструментирован с помощью afl-fuzzer (версия 4.32c).

Генерация содержимого файла `generated-seed`:

- Запустить исполняемый файл `seed`
- В ответ выведется машинозависимое число (возможно, каждый раз разное при нескольких запусках)
- Любое из выведенных чисел нужно сохранить в файл `generated-seed`

Fuzzing:

- Для определения какой-либо строки, на которой производёт падение требуется использовать утилиту afl-fuzz (можно ставить/настраивать локально, можно, и предпочтительно, использовать docker-образ версии 4.32 - на нём поведение проверено) `docker pull aflplusplus/aflplusplus:v4.32c`
- В файл answer нужно добавить строку, на которой приложение упадёт с segmentation faul
- В файл screen.png приложить вырезанный участок экрана со статистикой работы fuzzer (консольное окно, что выводится сразу полсе начала запуска, но с состоянием на момент нахождения ошибочной строки)

Сломать сервер

> Усложнённый вариант задания (при его выполнении первый можно не выполнять).

Последнее время в том числе начали набирать популярность REST-fuzzers, например:

- RESTler
- Cats
- ...

Суть задания в применении любого из средств REST API Fuzzing к тестовому сервису. Что

предполагается сделать:

- Спроектировать OpenAPI-спецификацию сервиса (может быть даже 1 метод) и выложить файл с именем test-openapi.json. Спецификация должна соответствовать стандарту с возможностью импортирования в стандартные средства типа Postman
- Реализовать сервис на python с кодом загрушки и любой исключительной ситуации. Код закоммитить с файлом requirements.txt для возможности установки / сборки
- Создать и выложить Dockerfile с образом настроенного выбранного средства и тестовым сервисом так, чтобы при запуске собранного контейнера сразу запускалось тестирование запущенного сервиса
- Сделать PR с пометкой в имени (REST-fuzzing)

Все наработки располагать в каталоге `rest-fuzzing`

Подсказки

AFL

- Заранее известно, что строка, на которой происходит падение приложений имеет длину не более 10 символов и не менее 1
- Также мы знаем, что параметр - текст