

# OSS-Fuzz 프로젝트를 위한 Continuous Fuzzing & Debugging 기술



지도교수님: 홍신 교수님

팀원: 김가륜, 이윤서, 조다빈, 김연희

## I. 문제 배경

### 연구 배경

OSS-Fuzz 는 850개 이상의 오픈소스 SW 프로젝트에 대해 지속적 인 테스트를 수행하는 프로젝트로 오픈소스SW 보안성 향상에 중요한 역 할을 하고 있다. 2016년부터 10,000개 이상의 Security vulnerability를 자동으로 탐지하는데 기여하고 2023년 현재 850개 이상의 오픈소스 SW 프로젝트를 대상으로 테스트를 수행되고 있다.

### 과제의 필요성

기존의 OSS-Fuzz 프로젝트는 프로젝트별로 개발자가 제공한 Fuzz Testing Driver 를통해서만 테스트를 수행하여 테스트를 통해 검사 할 수 있는 코드 커버리지가 한정적이었다. 최근에는 프로그램 분석, LLM 등을 통해 Fuzz Testing Driver 를 합성하여, 사용자의 입력 이 없이도 테스트를 수행할 수 있도록 자동화하는 기술들이 C/C++ 프로그램을 대상으로 소개되고 있다.

이에 본 프로젝트에서는 기존의 방법을 평가하고 개선하여 Java, Python, JavaScript에 대해 사용자가 입력 없이도 Fuzz Testing Driver를 자동으로 생성하여 OSS-Fuzz가 도달할 수 있는 코드 영역을 확장하는 자동화 기법을 개발하고자 한다.

## II. 문제 정의

### Problem Statement

- 기존 OSS-Fuzz의 Fuzz Testing Driver의 형태와 기능을 파악하고, Java, Python, JavaScript에 대해 사용자가 입력 없이 도 Fuzz Testing Driver를 자동으로 생성하는 신규 프로젝트를 개발하여 OSS-Fuzz가 도달할 수 있는 코드 영역을 확장한다.

### Constraints

- 본 프로젝트에서는 사용자의 입력을 받지 않고 Fuzz Testing Driver를 자동으로 생성해야 한다.

### Objectives

- 기존에 나온 자동화 기술들을 개선하여 Java, Python, JavaScript에 대해 적용하는 기법을 개발하는 것이 목표이다.

### Functions

- 코드 합성 기술을 통해 Java, Python, JavaScript 로 개발된 프 로젝트에 적합한 Fuzz Testing Driver 생성 기술

## III. 백그라운드 스터디

### 1. System Programming

- C언어를 통해 Fuzzing 학습 전 선행되어야 할 기초적인 System Programming 지식에 대해 복습했다.
- File I/O, Socket Programming, Multi-Process, Multi-Thread 등의 분야를 집중 학습했다.

### 2. Introduction to Software Testing

- Software Testing은 SW 개발 과정에서 중요한 단계로, 현재 많 은 부분이 자동화되어 있는 상태이다. Software Quality가 낮을 경우 발생할 수 있는 사회적/경제적 손실을 Testing을 통해 방지할 수 있다는 점에서 Testing의 중요성을 확인할 수 있다.
- Program Fault의 종류와 Fault to Failure의 필요충분 조건 인 Execution-Infection-Propagation(PIE)를 적용/분석 했다.

### 3. Structural Coverage Measurement

- Gcov는 gcc tool의 일종으로, 이를 통해 프로그램의 Line coverage와 Branch coverage를 측정할 수 있다.
- Clang의 Source-based coverage는 Compiler의 Front-end level에서 coverage를 구하는 Tool이다.

## IV. 접근 방법 및 설계

### 핵심 기술

- AFL++
  - code coverage를 효율적으로 늘리기 위해 genetic algorithm을 이용한 fuzzer
- LibFuzzer
  - 지금도 계속 발전하고 있는 coverage 기반의 fuzzing engine
  - test 아래의 library와 연결되어 있어서 target function 을 통해 library에 fuzzed input을 제공한다.
  - fuzzer는 cod coverage를 최대화 하기 위해서 입력 데이 터의 집합을 변화시킨다.
- OSS-Fuzz
  - 구글에서 제공하는 fuzzing service
  - 최신 fuzzing 기술과 확장 가능한 distributed execution을 결합해 OSS를 더욱 안전하고 안정적으로 만든 다.

