Fuzz Driver 합성

Tae Eun Kim

2022.05.05

Abstract

퍼징은 자동으로 생성된 수많은 입력을 통해 프로그램을 테스트하는 기법이다. 만약 퍼징으로 라이브러리의 결함을 찾고 싶다면 퍼즈 손잡이가 필요하다. 퍼즈 손잡이 작성은 복잡한 작업이라 주로 수작업으로 이루어져 왔다. 하지만 최근들어 자동화의 움직임이 불고 있다.

퍼징은 자동으로 생성된 수많은 입력을 통해 프로그램을 테스트하는 기법이다. 구글의 OSS Fuzz 프로젝트에 의하면, 지난 6년간 다양한 퍼저들은 3만5천 건의 결함을 발견했다. 이렇게 효과적인 퍼징이 가장 필요한 곳은 아마 오픈소스 라이브러리일 것이다. 왜냐하면 라이브러리에 있는 결함은 해당 라이브러리를 사용하는 다양한 프로그램들에게까지 영향을 미치기 때문이다. 하지만 라이브러리를 퍼징하는 것은 그리 단순하지 않다.

라이브러리 프로그램을 퍼징하기 위해선 퍼즈 손잡이(퍼즈 드라이버-Fuzz Driver라고도 한다)가 필요하다. 왜냐하면 라이브러리 프로그램은 그 자체로 실행될 수 없기 때문이다. 따라서 해당 라이브러리의 함수를 호출하여 사용해주는 프로그램이 있어야 라이브러리의 코드가 실행될 수 있다. 이 때, 퍼즈 손잡이는 라이브러리를 호출하는 작은 프로그램으로써, 퍼저와 라이브러리를 이어주는 역할을 한다.

제대로 된 퍼즈 손잡이는 올바르게 라이브러리를 사용하며, 동시에 라이브러리의 핵심적인 코드를 최대한 많이 실행해야 한다. 우선 올바르게 작성되어야 제대로 라이브러리의 기능을 실행할 수 있을 것이다. 또한 퍼즈 손잡이의 목적은 퍼징이므로, 퍼즈 드라이버를 통해 결함을 발견할 수 있어야 한다. 이 때, 핵심적인 코드를 최대한 많이 실행할 수록 결함을 발견할 확률도 높아진다. 따라서 제대로 된 퍼즈 손잡이를 작성하기 위해서는 라이브러리의 사용법과 내용을 잘 이해하고 있어야 한다. 그렇기 때문에 퍼즈 손잡이는 주로 수작업으로 작성되며, 이는 퍼징을 손쉽게 시행하기 어려운 이유가 된다.

최근 들어, 퍼즈 손잡이를 자동으로 합성하는 기술이 연구되고 있다. 아직까지는 라이브러리의 실제 사용 예제를 통해 올바르기만 한 퍼즈 손잡이를 만들어내고 있다. 하지만 여기에 데이터 흐름을 고려한다면 더 많은 라이브러리 코드를 실행할 수 있을 것이다. 그렇다면 올바르면서 효과적이기까지 한 퍼즈 손잡이를 자동으로 생성할 수 있을 것으로 기대된다.