Contract Fuzzer: Fuzzing Smart Contracts for Vulnerability Detection

# Introduction

암호화폐는 블록체인상의 서로 공유된 원장과 사용자들 간의 합의 프로토콜을 사용하여 신뢰할 수 없는 환경에서 안전한 거래를 가능하게 한다. 비트 코인이 개발된 이후, 많은 양의 블록체인 플랫폼이 개발되었는데, 그중 하나인 Ethereum은 Ethereum Virtual Machine을 사용하여 블록체인 위에서 거래 프로그램인 Smart Contract를 실행할 수 있게 한다. 대표적인 예시로 Etheroll은 자신의 Ether을 활용하여 도박을 할 수 있다. 그러므로 Smart Contract에서 취약점이 발견될 경우, 해당 사용자에게 큰 경제적 손실을 야기할 수 있다. 본 논문에서는 Smart Contract내에서 보안적인 취약점을 발견하는 Fuzzing Framework를 제안한다.

# Problem Definition

Smart Contract 개발자가 블록체인 플랫폼에 대한 부족한 이해로 취약점을 만들어 낼 수 있음

취약점을 미리 정의 후 EVM 로그를 분석하여 발견

Gas는 Ethereum 네트워크에서 컴퓨팅 자원을 사용하면 지불해야하는 세금으로 Smart Contract의 무한반복을 막음

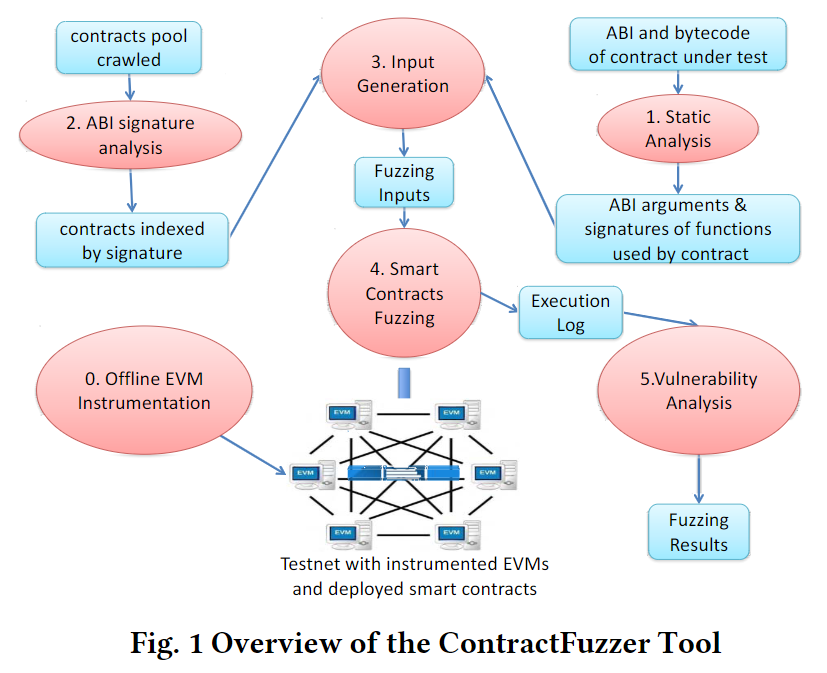
* + Gasless send
    - send함수를 사용할 때 gas를 쓰지 않아서 발생하는 예외처리를 제대로 핸들링 하지 않을 경우
  + Exception disorder
    - 함수마다 exception handling하는 방법이 달라서 발생
    - Exception 발생하면 모든 nested call을 전부 다 취소해야 됨
    - 다른 contract를 호출하는 call function의 경우 nested call을 더 이상 취소하지 않음
  + Timestamp/Block Number dependency
    - Ether을 보낼 때 블록체인 상의 저장된 정보를 조건문으로 사용할 때 발생
    - 두 정보의 경우 채굴자에게 의존적인 정보
  + Freezing ether
    - Ether를 보내는 기능을 외부의 Smart Contract에 의존할 때 외부의 smart contract를 종료하면 Ether 동결

Smart Contract verification tools가 이미 존재

* + Detection Strategy 부정확

# Main Idea

Contract Fuzzer를 제시

* + ABI Interface를 분석해서 fuzzing에 사용될 input을 generate
  + 정의된 취약점을 발견하는 테스트 오라클을 사용
  + E.g., log를 분석하여 TIMESTAMPOP 명령어와 ether을 보내는 send 명령어가 같이 발견되었을 경우 취약점 발견
* Contract Fuzzer Component

Contracts pool crawled

* + 웹크롤링 기술을 활용하여 배치된 smart contract 코드를 추출
  + 로컬 testnet 환경에 deploy

Static Analysis

* + Ethereum ABI는 JSON파일 형식으로 함수의 정보와 매개변수의 데이터 타입을 포함
  + 함수 이름과 입력에 필요한 매개변수 type을 매칭

ABI signature analysis

* + Ethereum의 바이트 코드는 함수이름을 keccak 해싱한 결과의 앞 4byte 값을 명시
  + 바이트 코드를 분석하여 같은 함수를 호출하는 Smart Contract들을 매핑

Input Generation

* + 특정 함수를 지원하는 Smart Contract에게 Data type에 맞게 Random Input을 생성

Smart Contracts Fuzzing

* + 함수 단위로 Fuzzing을 수행

Vulnerability Analysis

* + 임의로 정의된 Test oracle에 따라 EVM 실행로그를 분석하여 취약점 도출

# Evaluation

기존의 에러감지 툴인 Oyente verification tool과 비교

* + Oyente는 3개의 취약점만 발견할 수 있는데 Contract Fuzzer는 7개의 취약점을 발견
  + Oyente는 Smart Contract의 흐름을 코드기반으로 분석하여 취약점을 발견

TimeStamp Dependency

* + 텍스트, 화면, 스크린샷이(가) 표시된 사진

    자동 생성된 설명
  + SlotMachine contract 내부에서 타임스탬프를 ether을 보내는 조건으로 쓰는 것을 발견

# Outcomes

Smart Contract의 취약점을 발견하는 Fuzzing Framework를 구축

# Strengths

Fuzzer를 smart contract에 적용해서 효과적인 결과를 얻음

# Weaknesses

Smart Contract 내부의 취약점에 집중하여 EVM이나 블록체인 플랫폼에 대한 취약점은 고려하지 않음

|  |  |  |  |
| --- | --- | --- | --- |
| Authors | Bo Jiang | Ye Li | W.K. Chan |
| Affiliation | Beihang University | Beihang University | City University of Hongkong |
| Published at | ASE 2018 |  |  |
| Year | 2018 | Number of Citations | 265 |
| Time at Started | 18/3/2022 7:00 PM | Time at Completed | 25/3/2022 10:47 AM |

# Adaptation to Our Research

1. 공격에 활용될 수 있는 Smart contract의 취약점을 추가적으로 정의 후 발견