总结报告

ContractFuzzer：基于模糊测试的智能合约漏洞检测系统

常志伟[1]

1. 南开大学网络空间安全学院
E-mail: 2778571381@qq.com

1 研究背景

智能合约是运行在区块链共识协议之上的程序代码，其能够使人们在最小化彼此之间信任的同时达成协定。如今，数百万的智能合约被部署在各种各样的去中心化应用中，然而存在于智能合约中的安全漏洞却对这些去中心化应用造成了巨大的威胁。智能合约容易受到安全攻击的几个原因：

智能合约的运行依赖于底层的区块链平台以及其它协作合约，合约开发者没能完全理解这些合约之间以及合约与底层区块链平台之间隐含的关系；智能合约的编程语言与运行环境对于合约开发者来说都是全新的，且这些工具还不够成熟，合约开发者没能很好的处理这些工具自身的不足；区块链具有不可篡改的性质，智能合约在被部署至区块链平台之后难以更新。虽然以往的工作中提出了几种检测智能合约安全漏洞的验证工具，但这些工具依旧存在着一些局限性：

检测策略可能不精确，容易导致高误报率；符号验证所有可行的路径可能存在路径爆炸的问题，如果仅验证某些路径容易导致漏报。

本文提出了 ContractFuzzer 工具，该工具是一个全新的针对于以太坊智能合约安全漏洞检测的模糊测试工具，其能够根据智能合约的 ABI 规范生成模糊测试输入，定义检测安全漏洞的测试预言（test oracle），通过对以太坊虚拟机（EVM）插桩记录智能合约的运行时状态，分析日志并报告安全漏洞。

2 主要贡献

1）文章实现了第一个用于检测以太坊智能合约安全漏洞的模糊测试框架；

2）文章提出了一系列用于精确检测真实智能合约安全漏洞的测试预言（test oracle）；

3）文章系统地对以太坊平台上的 6991 个真实的智能合约进行了模糊测试，ContractFuzzer 工具发现了 459 个合约中存在的安全漏洞。

3 研究方法

3.1 以太坊智能合约

从概念上来说，以太坊平台可以被视为一个基于交易的状态机，其状态在每次交易时进行更新。交易的有效性由底层区块链平台的共识协议进行验证。基于区块链的去中心化应用中的安全漏洞可能存在于区块链层面、以太坊虚拟机（EVM）层面以及智能合约层面。文章的关注点主要集中在以太坊智能合约中的安全漏洞。主要对下面七种智能合约漏洞进行检测：

- Gasless Send
- Exception Disorder
- Reentrancy
- Timestamp Dependency
- Block Number Dependency
- Dangerous DelegateCall
- Freezing Ether

3.2 针对智能合约漏洞定义测试预言

3.2.1 Gasless Send.

- 测试预言 GaslessSend 检查：(1) 函数调用是通过 Send 函数进行的，即该函数调用的输入为 0，Gas 限制为 2300；(2) 函数调用在执行时返回错误码 ErrOutOfGas。

3.2.2 Exception Disorder.

- 对于一个嵌套调用链，测试预言 ExceptionDisorder 检查：由原始调用开始的嵌套调用链中的调用抛出异常，但是原始调用没有抛出异常。

3.2.3 Reentrancy.

测试预言 Reentrancy 包括两个子预言 ReentrancyCall 与 CallAgentWithValue：

$$Reentrancy = ReentrancyCall \quad CallAgentWithValue$$

- 子预言 ReentrancyCall 检查：原始函数调用在由其开始的嵌套调用链中出现了不止一次。
- 子预言 CallAgentWithValue 检查：(1) 函数调用所发送的以太币大于 0; (2) 被调用函数拥有充足的 Gas 执行复杂的代码，即函数调用不是通过 Send 函数或 Transfer 函数进行的；(3) 被调用合约由原始合约调用者指定，而不是硬编码在原始合约中的。

3.2.4 Timestamp Dependency Block Number Dependency

测试预言 TimestampDependency/BlockNumDependency 包括三个子预言 TimestampOp/BlockNumOp、SendCall 与 EtherTransfer：

TimestampDependency/BlockNumDependency = TimestampOp/BlockNumOp (SendCallEther-Transfer)

- 子预言 TimestampOp/BlockNumOp 检查：当前合约的执行过程中执行了 TIMESTAMP/NUMBER 操作符。
- 子预言 SendCall 检查：函数调用是通过 Send 函数进行。
- 子预言 EtherTransfer 检查函数调用所发送的以太币大于 0。

3.2.5 Dangerous DelegateCall

- 测试预言 DangerDelegateCall 检查：

当前合约执行过程中通过 DelegateCall 函数进行了函数调用；DelegateCall 函数的参数是由当前合约调用者指定的。

3.2.6 Freezing Ether

- 测试预言 FreezingEther 检查：

(1) 当前合约能够接收以太币；(2) 当前合约执行过程中通过 DelegateCall 函数进行了函数调用；(3) 当前合约自己的代码中没有 transfer/send/call/suicide 函数。

3.3 ContractFuzzer 系统架构

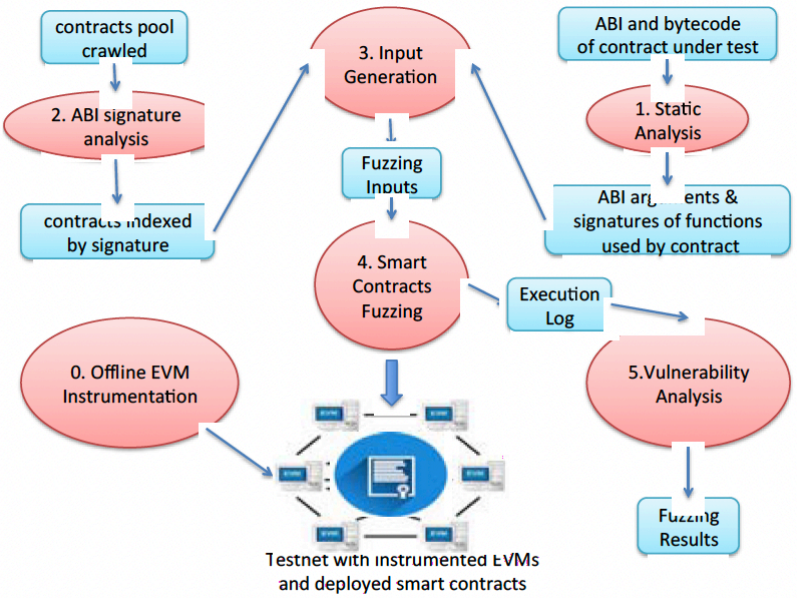

Fig. 1 Overview of the ContractFuzzer Tool

图 1 Main name 2

ContractFuzzer 工具包括一个线下的 EVM 插桩工具以及一个线上的模糊测试工具。线下的 EVM 插桩工具通过对 EVM 进行插桩，使得模糊测试工具能够监视智能合约的执行并提取执行日志用于漏洞分析。

文章实现了一个网络爬虫从 Etherscan 网站上爬取以太坊平台上已经部署的智能合约，爬取内容包括合约创建代码、ABI 接口与构造函数参数。文章将爬取的智能合约重新部署在自己搭建的以太坊测试网络中，一方面作为之后模糊测试的对象，另一方面作为使用合约地址作为参数的合约调用的输入。线上的模糊测试过程如下：

- 分析测试智能合约的 ABI 接口以及字节码，提取 ABI 函数的每一个参数的数据类型以及 ABI 函数中所使用到的函数签名；
- 对于所有从以太坊平台上爬取的智能合约进行 ABI 签名分析，并根据各个智能合约所支持的函数签名将其进行索引；
- 生成与 ABI 规范相符的合法模糊测试输入以及越过有效边界的突变输入；
- 启动模糊测试，通过随机的函数调用，使用生成的输入调用相应的 ABI 接口；
- 分析模糊测试过程中生成的执行日志，检测安全漏洞。

4 创新点

本文提出了 ContractFuzzer 工具，该工具是一个全新的针对于以太坊智能合约安全漏洞检测的模糊测试工具，其能够根据智能合约的 ABI 规范生成模糊测试输入，定义检测安全漏洞的测试预言（test oracle），通过对以太坊虚拟机（EVM）插桩记录智能合约的运行时状态，分析日志并报告安全漏洞。

5 总结

文章实现了第一个用于检测以太坊智能合约安全漏洞的模糊测试框架；文章提出了一系列用于精确检测真实智能合约安全漏洞的测试预言（test oracle）；文章系统地对以太坊平台上的 6991 个真实的智能合约进行了模糊测试，ContractFuzzer 工具发现了 459 个合约中存在的安全漏洞。