1. SmartGift: learning to generate practical inputs for testing smart contracts

Teng Zhou (1); Kui Liu (1, 2); Li Li (3); Zhe Liu (1); Klein, J. (4); Bissyande, T.F. (4)
Source: *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, p 23-34, 2021;
ISBN-13: 978-1-6654-2882-8; **DOI:** 10.1109/ICSME52107.2021.00009; **Conference:** 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME), 27 Sept.-1 Oct. 2021, Luxembourg City, Luxembourg;
Publisher: IEEE, Piscataway, NJ, USA
Author affiliation: (1) Nanjing University of Aeronautics and Astronautics, China (2) State Key Laboratory of Mathematical Engineering and Advanced Computing, China (3) Monash University, Melbourne, VIC, Australia (4) University of Luxembourg, Interdisciplinary Centre for Security, Reliability and Trust, Luxembourg
Abstract: With the boom of Initial Coin Offerings (ICO) in the financial markets, smart contracts have gained rapid popularity among consumers. Smart contract vulnerabilities however made them a prime target to malicious attacks that are leading to huge losses. The research community is thus applying various software engineering technologies to smart contracts to address them. In general, to detect vulnerabilities in smart contracts, mutation and fuzz based testing approaches have been widely studied and indeed achieved promising performance on benchmark datasets. Generating test inputs with mutation approaches essentially relies on the available test cases in a smart contract program. In our preliminary study, however, we observed that 56.4% of 218 identified open-source smart contract project repositories do not provide any test case for validation. Fuzzing test inputs leads to random values and lacks practical usefulness. Our work addresses this problem: we propose an approach, Smartgift, which generates practical inputs for testing smart contracts by learning from the transaction records of real-world smart contracts. Leveraging a collected set of over 60 thousand transaction records, Smartgift is able to generate relevant test inputs for ~77% smart contract functions, largely outperforming the traditional fuzzing approach (successful for only 60% functions). We further demonstrate the practicality of the test inputs by using them to replace the test inputs of the ContractFuzzer state of the art smart contract vulnerability detector: with inputs by Smartgift, ContractFuzzer can now detect 131 of the 154 vulnerabilities in its benchmark. (0 refs)
Inspec controlled terms: blockchains - contracts - fuzzy set theory - program testing - software engineering
Uncontrolled terms: mutation approaches - fuzz based testing approaches - open-source smart contract project repositories - relevant test inputs - smartgift - initial coin offerings - malicious attacks - software engineering technologies
Classification Code: C6150G Diagnostic, testing, debugging and evaluating systems - C6110B Software engineering techniques - C6130S Data security
IPC Code: G06F9/44 - G06F11/36 - G06F21/00 - H04L9/00
Treatment: Bibliography (BIB) - Practical (PRA)
Database: Inspec
Data Provider: Engineering Village