Smart Contract Tools Evaluation Papers.

Paper	Summary	Remark
1. ICSE 2020 Empirical Review of Automated Analysis Tools on 47,587 Ethereum Smart Contracts	• This paper analyses 9 existing smart contract evaluation tools. • The developed an environment called SmartBugs that includes the 9 existing tools for contracts evaluation. • In this paper, two different datasets are used to assess the efficacy of the existing tools from three aspects: Effectiveness, Production and Performance. • One of the datasets contains 69 contracts with known vulnerabilities and the second dataset contains 47,518 contracts collected from OSS repositories. • Its takes the authors about 564 days to run the tools on the datasets.	• Using the SmartBugs, it is easier to run the selected tools on the same dataset. • The dataset is available online • The time taken to evaluate the tools on the dataset is too long (about 1.5 years)
2. ISSTA 2021 Empirical Evaluation of Smart Contract Testing: What Is the Best Choice?	• This paper proposes four steps to evaluate existing smart contracts evaluation tools. • The authors first collect a dataset containing several vulnerability characteristics. • The dataset was evaluate on nine existing tools, 3 static analysis, 3 symbolic execution and 3 dynamic fuzzing tools. • The authors qualitatively analyses each tool based on the types of vulnerability they can detect from the dataset.	• The dataset is available online. • The paper focusses on having an environment with uniform parameters for fair comparison.
3. ISSTA 2020 How Effective are Smart Contract Analysis Tools? Evaluating Smart Contract Static Analysis Tools Using Bug Injection	• This paper propose a systematic approach to evaluate the smart contracts static analysis tools. • The authors presents SolidiFI an automated approach. • The authors injects bugs into contracts code and solidiFI test the generated bugs using static analysis tools to identify the bugs that the tools couldn't find. • The authors used 6 existing tools to run the SolidiFI. • The authors finds bugs that the existing tools couldn't detect despite their claims that they can detect such bugs.	• Data and code available : https://github.com/DependableSystemsLab/SolidiFI • Inject bugs to code to target the false negatives results
4. PRDC 2021 An Empirical Evaluation of the Effectiveness of Smart Contract Verification Tools	• This paper introduce a smart contract defect classification scheme based on the Orthogonal Defect Classification. • The authors apply this classification scheme to a contract dataset, which has been extracted from multiple sources and holds different types of defects. • Three tools were analyzed using the datasets to evaluate their performance on the fault detection.	• Introduce classification scheme for contracts bugs. • The nature of the bugs are not discussed in the paper

Open Questions

1. Several approaches are used to classify the vulnerability/bug, which of these classification schemes is the best?

    a. What are the existing bug's classification schemes for smart contract?
    b. What methods are used to classify the bugs?
    c. Which of the classification schemes best characterized the bugs?

2. Which class of vulnerability is more severe and which tool can detect it better?

    a. What are the bug's types?
    b. What are the characteristics of the severe bugs?
    c. Which of the existing tools can detect severe bugs?

3. To what extent the tools differs in reporting relevant vulnerabilities?

    a. What is the performance of the tools in detecting bugs?
    b. What types of bugs are the tools designed to detect?

Task Overview