Static Analysis of Ethereum Smart Contracts using Slither

Advanced Topics in Software Engineering

2232-CSE-6324-004

GitHub Link:

https://github.com/Akansha9812/CSE-6324-ADV-TOPS-SOFTWARE-ENGINEERING

Team - 7

Akansha Mohan - 1002031356

Sumanth Javvaji - 1001960021

Shree Tejaswini Kadali - 1002026757

Bhanuja Jannepalli - 1002028506

Overview

This course requires us to work with any tool which can be used to analyze source code, executables or even documentation to find problems before they cause a major security issue in a network. Slither, a static analysis tool is one of the tools which helps mitigate security issues by providing features like automated vulnerability, optimization detection as well as assistive codebase summaries to further developer comprehension [1].

Competitors of Slither

Static Analysis Tools

There are various Static Analysis Tools like Securify, SmartCheck, Solhint to detect problems but we have decided to use Slither because of the following reasons :

1. Accuracy - It is the most accurate tool with the lowest false positive rate of 10.9% [2].
2. Performance - Experiments deduced that slither was as fast as a simple linter [2].
3. Robustness - Slither failed for only 0.1 % of the contracts [2].

		Slither	Securify	SmartCheck	Solhint
Accuracy	False positives	10.9%	25%	73.6%	91.3%
	Flagged contracts	112	8	793	81
	Detections per contract	3.17	2.12	10.22	2.16
Performance	Average execution time	0.79 ± 1	41.4 ± 46.3	10.9 ± 7.14	0.95 ± 0.35
	Timed out analyses	0%	20.4%	4%	0%
Robustness	Failed analyses	0.1%	11.2%	10.22%	1.2%
Reentrancy examples	DAO	✓	✗	✓	✗
	Spankchain	✓	✗	✗	✗

Fig.1 [2]

Features

Slither can find vulnerabilities in smart contracts without user intervention. It provides various printers which allows it to print contract information in customized summary formats. As part of this project we are working upon below open issues related to Data dependency and Call graph printers [3].

1. **Issue #1127 : Data Dependency printer - Infinite loop issue** : As part of this feature, we are working upon a data dependency printer module,which prints the data dependency between variables. This feature is currently going into an infinite loop and our focus is to identify the bug in the module ,fix and test it. [4]

2. **Issue #664 : Call-graph printer - functional overloading issue** : As part of this feature, we are looking into Call-graph printers, which enlist the call-graph functions in the contract. Currently the feature is not distinguishing between the overloading functions, hence our focus is to fix it using signatures and test it with corresponding smart contracts [5].

Risks

Risks	Probability[6]	Impact of Risk	Solution
Need to understand about slither code files in detail.	40%	High	Working a couple of more hours to become familiar with its features.
Testing with limited smart contract	20%	Medium	Finding the relevant smart contract which has the target issues
Process of installing and testing system compatibility.	5%	Low	Using a virtual environment, if installation fails

Customers and Users

Smart Contract Developers

'They create architecture for existing or newly developed software solutions using blockchain platforms and smart contracts and can integrate them into existing solutions' [7].

Smart Contract Auditors

They have the authority to eliminate flaws which could be used to perform unauthorized activities by malicious actors. Their job is to provide a client with a report depicting all detected vulnerabilities and their severity level [8].

References

- [1] https://www.alchemy.com/dapps/slither , accessed 2/13/23.
- [2]https://blog.trailofbits.com/2019/05/27/slither-the-leading-static-analyzer-for-smart-contracts/ , accessed 2/13/23.
- [3] https://www.immunebytes.com/blog/slither-a-solidity-static-analyzer-for-smart-contracts/ , accessed 2/13/23.
- [4] https://github.com/crytic/slither/issues/1127 , accessed 2/13/23.
- [5] https://github.com/crytic/slither/issues/664 , accessed 2/13/23.
- [6]https://projectmanagementacademy.net/resources/blog/probability-of-occurrence/ , accessed 2/13/23.
- [7] https://boostylabs.com/smart-contract , accessed 2/13/23.
- [8] https://hacken.io/services/blockchain-security/smart-contract-security-audit/, accessed 2/13/23.