INCEPTION PRESENTATION

Abhishek Wadhwani – 10020352719
Mounika Kottapalli– 1002085510
Nitin Raj Thumma– 1002080555
Sai Raghu Rami Reddy Dontireddy – 1002014523

GitHub: https://github.com/Abhismoothie/Slither-Enhancement-project-team-9-CSE6324-001

TABLE OF CONTENTS

S .ITHER

Slither is a **tool** that helps developers to analyze their Solidity code. It can detect **vulnerabilities** in the code and provide **visual information** about the contract details. Slither also allows developers to write their own custom analyses.

By using Slither, developers can **improve their code quality** and **find vulnerabilities** in their code more easily. It also helps in identifying where the error condition occurs in the source code. Most Importantly its average execution time is less than 1 second per contract.

FEATURES OF SLITHER

Slither uses Python, most popular and widely-range programming language, which making it accessible to developers.

It can detect a large amount of variety smart contract bugs which can be detected without the user intervention

Slither can even detect the code optimization which even the compiler misses.

FEATURES OF SLITHER

In Slither printers gives the summarize of contract’s information and display it so that it helps in study of codebase

Through Slither’s API a user can easily interact with it.

USING SLITHER

To install you need to have pyhton installed on your system with 3.6 or higher and then use command prompt to run the script.
pip3 install slither-analyzer

In command prompt run the command.
slither init MyProject
This command will create a project structure with the necessary files and directories.

Inside the project directory, there will "contracts" folder. We need to create a smart contract file inside this folder with **.sol extension**
After opening the file, we can write a smart contract in solidity

To compile we use the below code
slither compile
Brownie will compile your contract and generate JSON files with contract metadata in the "build" directory.

USING SLITHER

To create a test files for the smart contract we need to Create test files in the "tests" directory to write tests for your smart contract.
These test files are in python.

After creating the test files, we need to run the command on the command prompt.
slither test

LIMITATIONS OF SLITHER

False Positives and False Negatives : Slither can sometimes produce false positives, indicating potential vulnerabilities that don't actually exist in the code.

Limited Coverage : Slither may not cover every possible security issue or coding best practice. It focuses on a set of predefined rules and may not detect novel vulnerabilities or patterns that are not included in its rule set.

No Fix Suggestions : Slither provides information about potential issues but does not offer automatic fixes or code suggestions.

LIMITATIONS OF SLITHER

Lack of Context : Slither analyzes code statically, which means it doesn't consider runtime behavior or interactions with other smart contracts. This can limit its ability to detect certain vulnerabilities that only manifest during contract execution.

Outdated Rules : Slither's rule set may become outdated as new Solidity features, best practices, and security considerations emerge. Users need to ensure they have the latest version of Slither and regularly update their security analysis tools

PROJECT ENHANCEMENT

New Detector - Foundry Expect emit

In foundry tests, the expectEmit feature accepts multiple arguments including 4 Booleans. It is easy to make a mistake. This detector will ensure that the arguments correctly match the event signature.
For Example:

```
// event MyEvent(uint data, address indexed addr1, address
indexed addr2, address indexed addr3);
vm.expectEmit(true, true, false, true);
emit MyEvent(1, addr1, addr2, addr3);
myContract.callFunctionThatEmitsMyEvent();
```

PROJECT ENHANCEMENT

False – Positive Identifier

Slither’s naming – convention detector strictly picks up any names which are not camelCase or PascalCase. There is a convention that is becoming increasingly more popular where by the prefix i_ is used for immutables and s_ is used for state variables.

Slither should not report those cases for this detector. It occurs very Frequently.

REFRENCES

https://github.com/crytic/slither/issues/2083

https://github.com/crytic/slither/issues/2036

https://agroce.github.io/wetseb19.pdf

https://github.com/crytic/slither

QUESTIONS?

THANK YOU!