Final Report

Adding new detectors to Slither

The University of Texas at Arlington

Advanced Topics in Software Engineering

Fall 2022, CSE 6324 - 001

GitHub link: https://github.com/AtulUpadhye17/CSE-6324-Team4

Team 4:

Atul Upadhye (1002030159)

Kundana Vaka(1001827398)

Yamini Dhulipalla(1001913007)

Srikar Sai Yarlagadda(1001860709)

Vigneshwar Selvaraj(1001863627)

Project Plan:

Objective:

For our project, we plan to implement new detectors and integrate them into the slither developer version to analyze the solidity smart contracts efficiently.

Overview of Slither differs from other static analysis tools:

"Trail of Bits" has published a paper on Slither and compares its bug detection with other static analysis tools by doing experiments for finding issues in Ethereum smart contracts in terms of speed, robustness, the balance of detection, and false positives.[1]

Comparison with Existing tool for Iteration 4:

We compared Slither with other static analysis frameworks such as Solhint, Ethlint :

Detectors	Solhint	Ethlint	Current Slither	Extended Slither (Our Tool)
Incorrect Ordering	No	No	No	Yes

Iteration 4 Plan:

- Merge all the ordering-related detector codes into one detector.
- Optimized the search criteria for the " Imports on the Top" and "Pragmas on the Top" detectors.
- Implement and integrate the detector.
- Test those detectors and check for any bugs and fix them.
- Create the documentation of the project.

Risks Faced during Iteration 4:

Risk	Type	Risk Exposure	Plan for Mitigating
Complications while integrating all the ordering-related detectors (Function, contract, library) into a single detector. It was time-consuming to fix the errors in this process.	Major	This particular risk has a 80% chance of happening, and it will take 9 hours to fix it in this iteration. The risk exposure for this risk is therefore 7.5 additional hours of work resolving it.	Modified all the detector functions and modules in line with the slither API functions which made the integration less challenging.
Finding appropriate sample contracts for testing the detectors we have developed and integrated into slither was difficult.	Major	This particular risk has a 75% chance of happening, and it will take 10 hours to fix it in this iteration. The risk exposure for this risk is therefore 7.5 additional hours of work resolving it.	We have asked Shovon to help us in finding the appropriate sample contracts for testing the detectors.

Possible Risks in the future:

Risk	Type	Future Risk Exposure	Plan for Mitigating
Testing various smart contract samples in the future with developed detectors might result in different errors which need to be resolved	Major	This particular risk has 60% chance of happening, and it might take 10 hours to fix it. The risk exposure for this particular risk is therefore 6 additional hours of work resolving it.	Continue testing the tool with more smart contract samples in the future.
With blockchain growing it's possible that new vulnerabilities may be found, and we'll need to modify the slither tool to find those vulnerabilities	Major	This particular risk has 50% chance of happening, and it might take 10 hours to fix it. The risk exposure for this particular risk is therefore 5 additional hours of work resolving it.	Try to develop new detectors for finding vulnerabilities and integrate into slither or improving the existing detectors to find the newly encountered vulnerabilities.

Implemented Detectors:

Incorrect Ordering:

Description: Check the order of elements in the file and inside each contract, according to the style guide[7]

- Incorrect constructor order
- State variable declaration after function
- Library after contract
- Interface after library

Detector Code: Incorrect Ordering :

```
18      def _detect(self):
19          results = []
20          elements_list = []
21          for n in self.slither.crytic_compile.filenames:
22              rootFileName = n.absolute
23              with open(rootFileName, 'r') as f:
24                  for line in f:
25                      elements_list.append(line)
26
27              # check if the order of the constructor is correct or not
28              for contract in self.slither.contracts_derived:
29
30                  list_of_methods=contract.functions
31
32                  for x in range(len(list_of_methods)):
33                      #Check if constructor is present and placed before the functions
34                      if str(list_of_methods[x]).lower == "constructor" and list_of_methods[x]!=list_of_methods[0]:
35
36                          #info to be printed
37                          info = ['Incorrect constructor Order found in ',contract,"\n"]
38
39                          res = self.generate_result(info)
40
41                          results.append(res)
42
```

```python
        # check if the order of elements is correct or not
        for contract in self.slither.contracts_derived:
            for function in contract.functions:
                for element in elements_list:
                    if function.name in element:
                        if element.index(function.name) > elements_list.index(element):
                            info = ["The function {} is declared before the variable declaration.".format(function.name),"\n"]
                            res = self.generate_result(info)
                            results.append(res)

        # check if the library is declared before the contract or not
        # if the library is declared after the contract, print the line number and the error message.
        library_index = 0
        contract_index = 0

        for i in range(len(elements_list)):
            if elements_list[i].startswith("contract"):
                contract_index = i
            if elements_list[i].startswith("library"):
                library_index = i
        if library_index > contract_index:
            info = [f"The library is declared after the contract.","\n"]
            res = self.generate_result(info)
            results.append(res)

            # check if the interface is declared after the library or not
            # if the interface is declared before the library, print the line number and the error message.

            interface_index = 0
            library_index = 0

            for i in range(len(elements_list)):
                if elements_list[i].startswith("library"):
                    library_index = i
                if elements_list[i].startswith("interface"):
                    interface_index = i

            if interface_index < library_index:
                info = ["The interface is declared before the library.","\n"]
                res = self.generate_result(info)
                results.append(res)
        return results
```

Contract Sample:

```solidity
// SPDX-License-Identifier: MIT

contract MyContract {
    function hello() public{

    }
    //pragma
    constructor(){
    }


}

library MyLibrary {}

interface MyInterface {}

pragma solidity ^0.8.16;
```

incorrect_example.sol

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.16;
contract MyContract {
    function hello() public{

    }
    //pragma
    constructor(){
    }


}

library MyLibrary {}

interface MyInterface {}
```

Output Left Contract:

```
(slither-dev) [11/27/22]seed@VM:~/ordering$ slither incorrect_example.sol

solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Incorrect constructor Order found in MyContract (incorrect_example.sol#4-13)
Reference: https://github.com/trailofbits/slither/wiki/Adding-a-new-detector

The function hello is declared before the variable declaration.
The library is declared after the contract.
Reference: https://github.com/trailofbits/slither/wiki/Adding-a-new-detector

pragma statement should be on top : /home/seed/ordering/incorrect_example.sol
Reference: https://github.com/trailofbits/slither/wiki/Adding-a-new-detector
incorrect_example.sol analyzed (3 contracts with 86 detectors), 5 result(s) found
[2]+  Done                    gedit incorrect_example.sol
(slither-dev) [11/27/22]seed@VM:~/ordering$
```

Output Right Contract:

```
(slither-dev) [11/27/22]seed@VM:~/ordering$ slither incorrect_example.sol

solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Incorrect constructor Order found in MyContract (incorrect_example.sol#3-12)
Reference: https://github.com/trailofbits/slither/wiki/Adding-a-new-detector

The function hello is declared before the variable declaration.
The library is declared after the contract.
Reference: https://github.com/trailofbits/slither/wiki/Adding-a-new-detector
incorrect_example.sol analyzed (3 contracts with 86 detectors), 4 result(s) found
(slither-dev) [11/27/22]seed@VM:~/ordering$
```

Integration :

You can integrate your detector into Slither by:

- Adding it in slither/detectors/all_detectors.py.(We followed this approach)
- or, by creating a plugin package.

Integrated Detectors :

```
89 from .constructor.incorrect_constructor_name import IncorrectConstructorName
90 from .constructor.incorrect_constructor_order import IncorrectConstructorOrder
91 from .style.imports_on_top import ImportsOnTop
92 from .style.pragmas_on_top import PragmasOnTop
93 from .ordering.incorrect_ordering import IncorrectOrdering
```

Customers and Users:

- Any security audit firm, smart contract developer ,security expert, or academic researcher can use this tool with our new detectors added to Slither to make the smart contract auditing process more efficient.

Feedback:

- Shovon shared a few sample contracts with us and asked us to test them and verify if new detectors integrated into Slither can detect issues.

References:

[1] Overview. (n.d.). Retrieved October 15, 2022, from https://blog.trailofbits.com/2019/05/27/slither-the-leading-static-analyzer-for-smart-contracts/
[2] GitHub - crytic/slither: Developer installation. (n.d.). GitHub. Retrieved October 15, 2022, from https://github.com/crytic/slither/wiki/Developer-installation
[3] Rule Index of Solhint. (n.d.). Solhint. Retrieved October 17, 2022, from https://protofire.github.io/solhint/docs/rules.html
[4] User Guide — Solium 1.0.0 documentation. (n.d.). https://ethlint.readthedocs.io/en/latest/user-guide.html
[5] Adding a new detector · crytic/slither Wiki. (n.d.-b). GitHub. https://github.com/crytic/slither/wiki/Adding-a-new-detector
[6] imports-on-top Solhint. (n.d.). Solhint. https://protofire.github.io/solhint/docs/rules/order/imports-on-top.html
[7] Style Guide — Solidity 0.8.17 documentation. (n.d.). https://docs.soliditylang.org/en/v0.8.17/style-guide.html