CSE 6234 - ADVANCED TOPICS IN SOFTWARE ENGINEERING

Vulnerability Analysis using Slither tool

ITERATION 1

PRESENTED BY : TEAM 4

Preeti Singh - 1002013566
Sripal Thorupunoori - 1001969001
Mohit Singhi - 1002004892
Siddhartha Reddy Sungomula - 1001969005

5TLvT8h7NdqSRbcqqfNLvRDxl36O2qjDbd3uYBPRR15HG9P9VlvJJNe1o4ba4k-fQ8T2QxA_wBmQU2fiYAudj5oBYrNU1K0XlsQB6A6N3vqlEL_7r_DM97v-i4Yp0vfim0052kba

Project Plan

	Goal Targeted	Goal Achieved
Iteration #1	❖ Install Slither and understand the tool. ❖ Analyze the most closely related tools. ❖ Investigate/understand the issue in detail and find out what inputs, outputs, and data structures were used in detail. ❖ Identify and provide the necessary test cases for the important code elements. ❖ Assess the major risks and devise realistic plans to mitigate them. ❖ Examine how the project's agenda impacts the user.	❖ Installed the Slither and understood the tool. ❖ Analyzed the competitors like Mythril, Manticore. ❖ Investigated the alias issue in detail and found out what inputs, outputs, transition graph, and data structures. Identified the important code elements and written the necessary test cases for the same. ❖ Assessed the major risks and made necessary realistic plans to mitigate them. ❖ Examined how the support to import with alias will impact the user.

https://www.vectorstock.com/royalty-free-vector/project-planning-thin-line-icon-sign-symbol-vector-20702986

Project Plan

	Goal Targeted	Goal Achieved
Iteration #2	❖ Start implementing "support for alias import" ❖ Reevaluate the risks and prepare a risk mitigation plan. ❖ Conduct user testing and obtain appropriate user feedback.	
Iteration #3	❖ Ensure that all deliverables are completed on time. ❖ Test the important code elements with the necessary test cases. ❖ Get feedback from users on the fix.	

https://www.vectorstock.com/royalty-free-vector/project-planning-thin-line-icon-sign-symbol-vector-20702986

Issue #1452 Inputs -

Counter.sol

14 lines (11 sloc) | 258 Bytes

```
1  // SPDX-License-Identifier: UNLICENSED
2  pragma solidity ^0.8.13;
3
4  contract Counter {
5      uint256 public number;
6
7      function setNumber(uint256 newNumber) public {
8          number = newNumber;
9      }
10
11     function increment() public {
12         number++;
13     }
14 }
```

Importer.sol

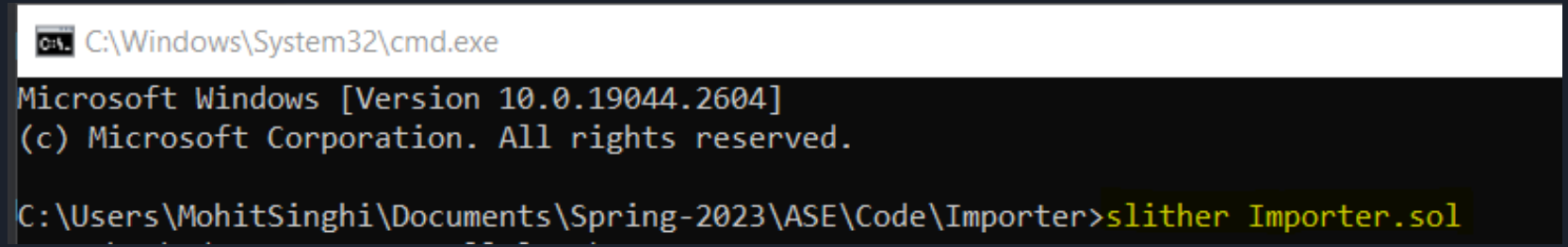
11 lines (8 sloc) | 171 Bytes

```
1
2  // SPDX-License-Identifier: UNLICENSED
3  pragma solidity ^0.8.13;
4
5  import "src/Counter.sol" as c;
6
7  contract Importer {
8      constructor() {
9          new c.Counter();
10     }
11 }
```


https://github.com/SheldonHolmgren/slither_bug_example/tree/master/src

Issue #1452 Outputs -

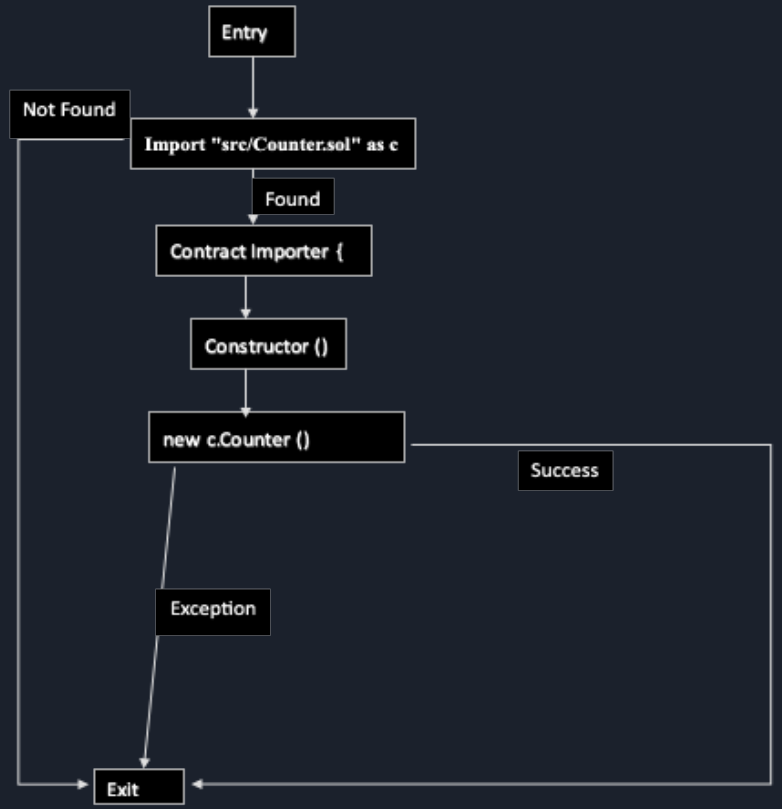
Recreation of Issue using slither in Real time



Error Stack Trace

```
    func.generate_slithir_and_analyze()
  File "C:\Users\MohitSinghi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-
packages\Python310\site-packages\slither\core\declarations\function.py", line 1750, in generate_slithir_and_analyze
    node.slithir_generation()
  File "C:\Users\MohitSinghi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-
packages\Python310\site-packages\slither\core\cfg\node.py", line 719, in slithir_generation
    self._irs = convert_expression(expression, self)
  File "C:\Users\MohitSinghi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-
packages\Python310\site-packages\slither\slithir\convert.py", line 119, in convert_expression
    result = apply_ir_heuristics(result, node)
  File "C:\Users\MohitSinghi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-
packages\Python310\site-packages\slither\slithir\convert.py", line 1858, in apply_ir_heuristics
    irs = propagate_type_and_convert_call(irs, node)
  File "C:\Users\MohitSinghi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-
packages\Python310\site-packages\slither\slithir\convert.py", line 442, in propagate_type_and_convert_call
    new_ins = propagate_types(ins, node)
  File "C:\Users\MohitSinghi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-
packages\Python310\site-packages\slither\slithir\convert.py", line 774, in propagate_types
    ir.lvalue.set_type(UserDefinedType(contract))
  File "C:\Users\MohitSinghi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-
packages\Python310\site-packages\slither\core\solidity_types\user_defined_type.py", line 19, in __init__
    assert isinstance(t, (Contract, Enum, Structure))
AssertionError
```

Transition Graph



Key Data Structures

- Abstract Syntax Tree
- Control Flow Graph
- Call Graph
- Data Flow Diagram
- Symbol Table

https://play-lh.googleusercontent.com/a4Xrc-8oQLu05mOrNPuvA_o2nZEIEnOoTH4wB91Slw_hCvulu_Qqi440bK9mC8ml-KA=w480-h960

Test Case	T1	
Description	Creating a new contract Vulnerable.sol which has vulnerability in it. Such that it will cause unchecked return value problem	
Input	**// Vulnerable.sol** pragma solidity ^0.8.0; contract Vulnerable { function transferTokens(address to, uint256 amount) external { (bool success,) = to.call{value: amount}(""); require(success, "Transfer failed"); } }	**// Importer.sol** pragma solidity ^0.8.13; import "src/Vulnerable.sol"; contract Importer { constructor() { new Vulnerable(); } }
Expected Output	When the Vulnerable.sol contract is imported instead of the expected Counter.sol contract, slither must notice a problem and deliver a warning sign.	

Risk management

Biggest Risks :

1. Delay in completion of the iterations.
2. The tool does not run consistently on all operating system.
3. Unknown issues may arise while solving the current problem.
4. Limited understanding of the tool.
5. Absence of a team member due to emergency.

https://static.vecteezy.com/system/resources/thumbnails/016/389/950/small/business-risk-management-free-png.png

Impact of risks:

Sl no	Description	Risk Exposure
Risk 1:	Delay in completion of one iteration can have impact on the next iteration, which delays overall project.	P= 20% E=20; R.E= 4 hours
Risk 2:	Risks like this can impact because it becomes it consumes a lot of time in setting up virtual machine with compatible OS	P= 40% E=20; R.E= 8 hours
Risk 3:	These kinds of risks have a high impact as unknown error makes the problem solving more complex.	P= 20% E=40; R.E= 8 hours
Risk 4:	Having limited knowledge requires the team to spend additional hours to gain knowledge on the issue itself.	P= 40% E=40, R.E= 16 hours
Risk 5:	Team members absence does impact the progress and productivity	P= 5% E=10; R.E= 0.5 hours

Plans to mitigate the risks

Risk 1: To deal with such cases is to have regular meetings on the work we have done.

Risk 2: To handle such risks is to set up a virtual machine on the operating system that can support the tool.

Risk 3: To mitigate risks from unknown errors we need to keep the tools up to date and follow best practices.

Risk 4: To avoid this risk we need to work extra hours to understand the tool. Seeking help from online forums and product documents.

Risk 5: In such emergency cases, the team needs to divide the work among the remaining teammates so that it does not affect the team's efficiency.

Actual benefits for users & customers

- Users are more likely to analyze Solidity code containing imports with aliases more accurately and efficiently.
- Smart contracts can be made more secure and exploitable vulnerabilities can be reduced in production.

Competitors [2]

- Slither is the most accurate tool with the lowest false positive rate
- Slither can detect more types of vulnerabilities, such as reentrancy, floating point precision errors, and uninitialized storage variables, than some other tools.
- A variety of formats are supported, including JSON, CSV, and plain text.
- Detected issues are explained in a clear and concise manner, with recommendations on how to resolve them.

		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	✓	✗	✗	✗

[1]

References

[1]https://arxiv.org/pdf/1908.09878.pdf

[2]https://blog.trailofbits.com/2019/05/27/slither-the-leading-static-analyzer-for-smart-contracts/

[3] [Slither Bug #1452] https://github.com/crytic/slither/issues/1452

[4] https://github.com/SheldonHolmgren/slither_bug_example/tree/master/src

Questions ?!