ASE 6234 - ADVANCED TOPICS IN SOFTWARE ENGINEERING

A STATIC ANALYSIS FRAMEWORK FOR ETHEREUM

SLITHER

PRESENTED BY : TEAM 4

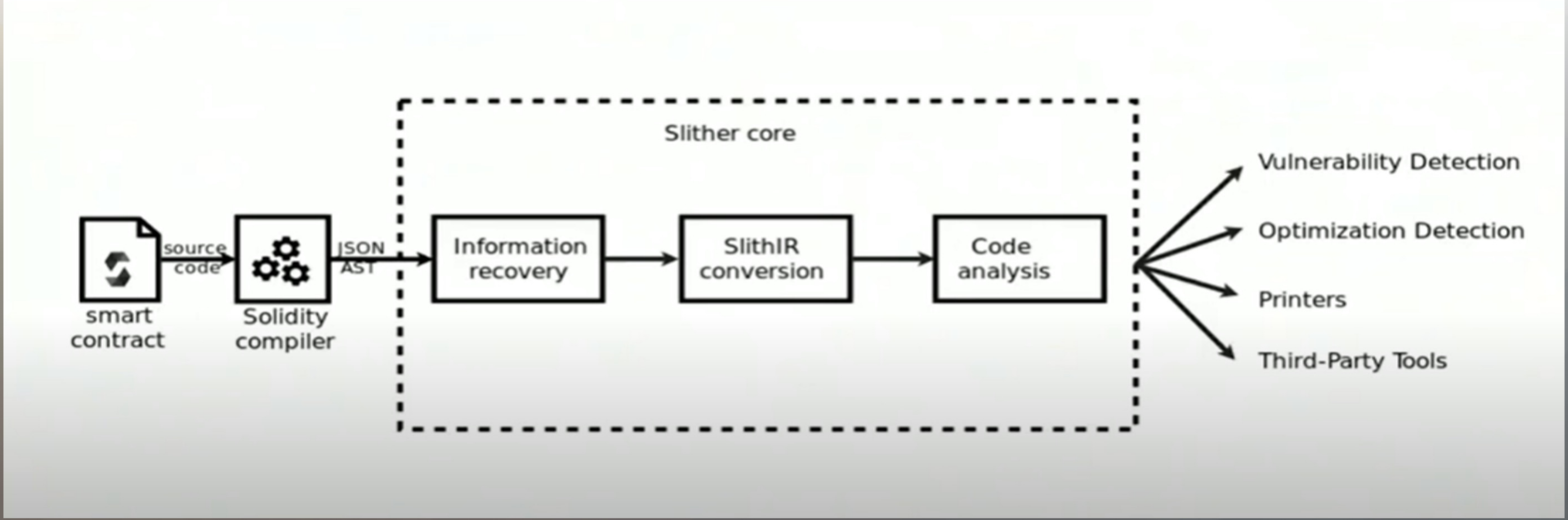
Preeti Singh	- 1002013566
Sripal Thorupunoori	- 1001969001
Mohit Singhi	- 1002004892
Siddhartha Reddy Sungomula	- 1001969005

OUTLINE

- ARCHITECTURE
- FEATURES
- A COMPARISON
- PROPOSED BUG FIXES
- SLITHER’S TARGET USER/CUSTOMER
- POSSIBLE RISKS
- COMPETITORS

Architecture : [1]

➔ Parsing ⟶ Analysis ⟶ Reporting



Features of Slither: [2]

- Vulnerability Detection
- Control Flow Analysis
- Custom Rule Creation
- Call Graph Analysis

Reference: features-list-icon-suitable-info-260nw-358530557.jpg

Proposed Bug Fixes to Slither:

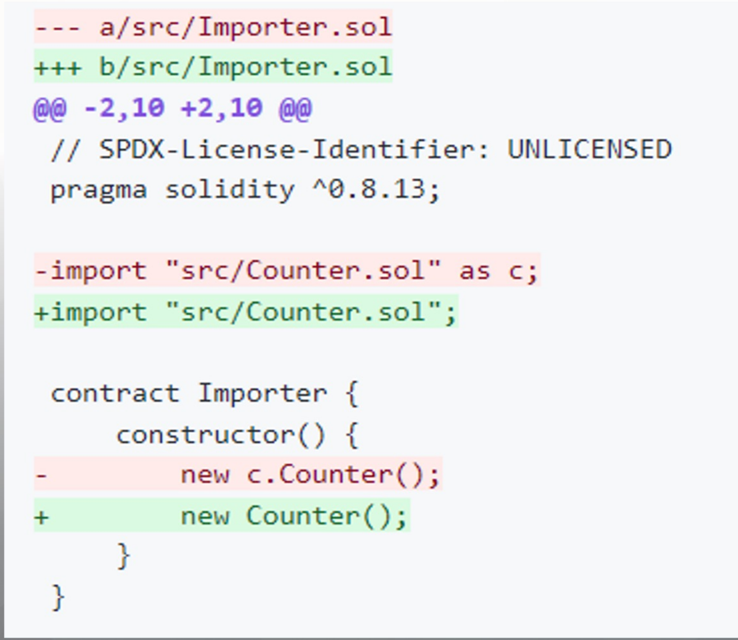
❖ Support for import with Alias #1452 [5]

- ➔ **When**: A external sol module is imported with Alias Name and objects inside that module are referred from alias.

- ➔ **Where**: The issue comes in Slither's *user_defined_type.py*

- ➔ **Impact:** High - This leads to a crash in Slither and it could not provide any analysis.

Sample Code Snippet:



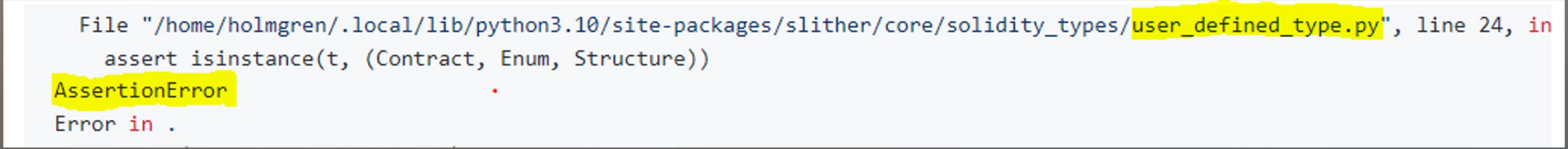
```
--- a/src/Importer.sol
+++ b/src/Importer.sol
@@ -2,10 +2,10 @@
 // SPDX-License-Identifier: UNLICENSED
 pragma solidity ^0.8.13;

-import "src/Counter.sol" as c;
+import "src/Counter.sol";

 contract Importer {
     constructor() {
-        new c.Counter();
+        new Counter();
     }
 }
```

[Slither Bug #1452]

Snip of Error Trace:



```
  File "/home/holmgren/.local/lib/python3.10/site-packages/slither/core/solidity_types/user_defined_type.py", line 24, in
    assert isinstance(t, (Contract, Enum, Structure))
AssertionError
Error in .
```

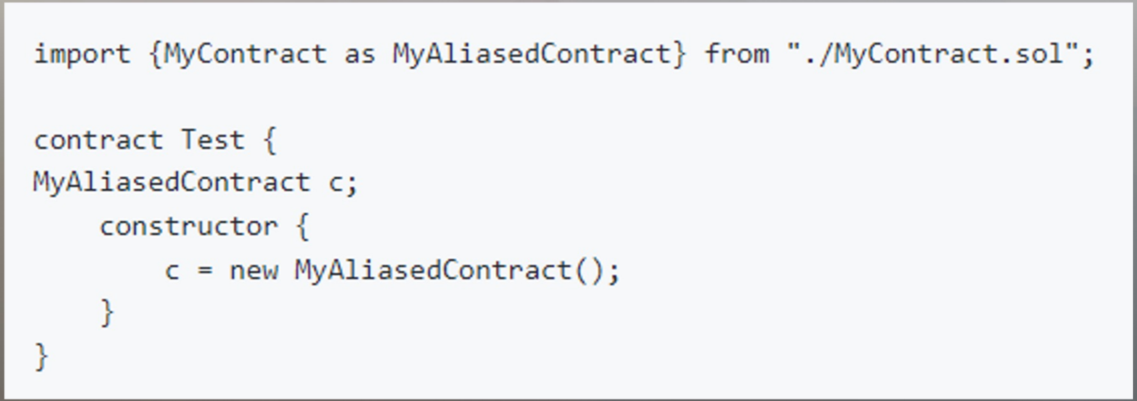
[Slither Bug #1452]

Related Issue:

❖ Import with alias collision #1364 [6]

When: An import renames a contract to a name that is already taken by another contract.

Impact: High - It causes Slither to crash.



```
import {MyContract as MyAliasedContract} from "./MyContract.sol";

contract Test {
MyAliasedContract c;
    constructor {
        c = new MyAliasedContract();
    }
}
```

[Slither Bug #1592]

Beneficial to (Targeted Users) :

- Smart Contract Developers
- Security Experts
- Auditors

Reference:105-1053050_continuous-improvement-continuous-improvement-icon.png

Possible Risks :

- Incorrect Analysis Result at times & Difficulty in Debugging
- Insufficient domain knowledge of the team (Solidity and Slither Vulnerability Detectors). May require up to 10 hours from each member to understand and get well-versed.

Risk Exposure:
Probability is 80%
Effort in hours = 10 * 4 (4 team members)

R.E. = P% * E
R. E. => 80/100 * 40
=> **32 hours** (extra to mitigate risk)

Competitors: [4]

➔ **Vulnerability Detection Evaluation - Re-entrancy Detectors**

- Securify, SolHint & SmartCheck

➔ **Optimization Detection Evaluation**

- Only Slither is Capable

➔ **Code Understanding Comparison**

- Surya tool

➔ **Threats to Validity**

- SmartAnvil

GitHub

https://github.com/preetisingh1121/Slither.git

References :

- [2][3] https://www.immunebytes.com/blog/slither-a-solidity-static-analyzer-for-smart-contracts/#How_Does_Slither_Work, accessed 02/10/2023
- [1][4] https://ieeexplore.ieee.org/document/8823898, accessed 02/11/2023
- [5] [Slither Bug #1452] https://github.com/crytic/slither/issues/1452 . accessed 02/13/2023
- [6] https://github.com/crytic/slither/issues/1364 . accessed 02/13/2023
- [Slither Bug #1592] https://github.com/crytic/slither/issues/1594 . accessed 02/13/2023