Advanced Topics in Software Engineering (CSE 6324)

Slither - A Static Analysis tool for Ethereum Smart Contracts

Team - 7

- Akansha Mohan (1002031356)
- Sumanth Javvaji (1001960021)
- Shree Tejaswini Kadali (1002026757)
- Bhanuja Jannepalli (1002028506)

Resources used

- Tool: Visual Studio COde v1.69.1.
- Repository: GitHub.
- Solc C v0.8.18 compiler.
- Python Z3 v4.5.0
- Docker.
- Slither.
- Solidity v0.8.1 programming language.

A few important terms

- ➔ Ethereum
- ➔ Solidity
- ➔ Smart Contracts

Is Ethereum secure?

Are you talking about the security of Ethereum or the security of Smart Contracts build on top of Ethereum

Tip

Vulnerabilities in the smart contracts. [1]

What is Slither?

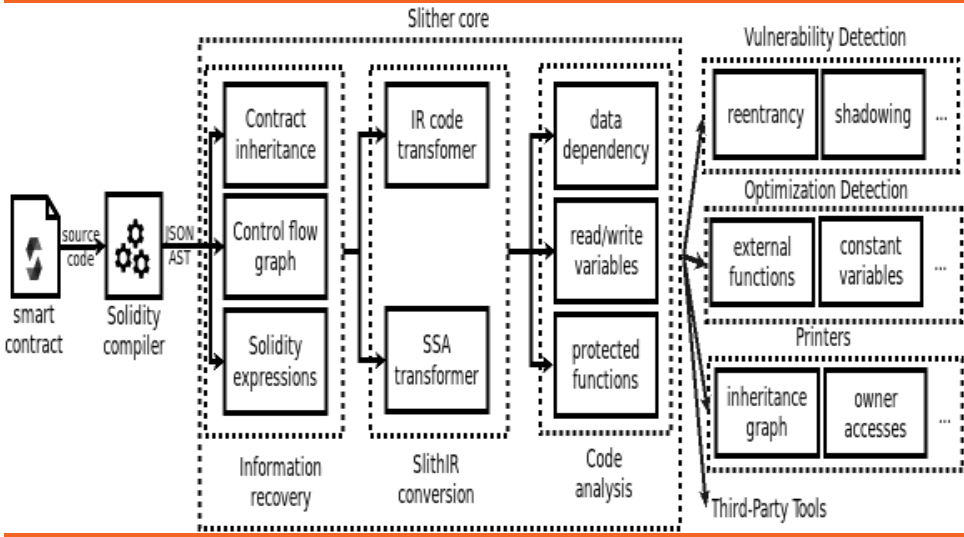
- A tool which helps to detect issues in the code without having to execute it. [2]
- Provides granular information about the code and flexibility necessary to produce many applications. [3]

Uses of static analysis framework, Slither [4]

Applications

- Automated Vulnerability Detection.
- Automated Detection of Code Optimization.
- Improved user's understanding of Contracts.
- Code review assistance.

Working…[5]


Slither core
Vulnerability Detection
reentrancy
shadowing
...
Optimization Detection
external functions
constant variables
...
Printers
inheritance graph
owner accesses
...
Third-Party Tools
smart contract
source code
Solidity compiler
JSON AST
Contract inheritance
Control flow graph
Solidity expressions
Information recovery
IR code transfomer
SSA transformer
SlithIR conversion
data dependency
read/write variables
protected functions
Code analysis

Competitors [11]

Securify, SmartCheck and Solhint

Accuracy, Performance and Robustness of Slither outperforms the rest by 10.9%, 0.79 and 0.1% respectively.

Customers

Smart Contract Auditors

Smart Contract Developers

Our Aim

Issue 1

To fix the infinite loop problem in data dependency. [6]

Issue 2

To fix functional overloading in call-graph printer. [9]

Issue 1 [7]

Effects

- Freezing of the network.
- Draining of resources.
- Security vulnerabilities.

Approach to fix problem 1 [8]

1. Use Slither to generate a report.
2. Identify the location of the infinite loop.
3. Understand the source of the infinite loop.
4. Fix it.
5. Test the fix.
6. Use Slither to check if the problem has been solved or not.

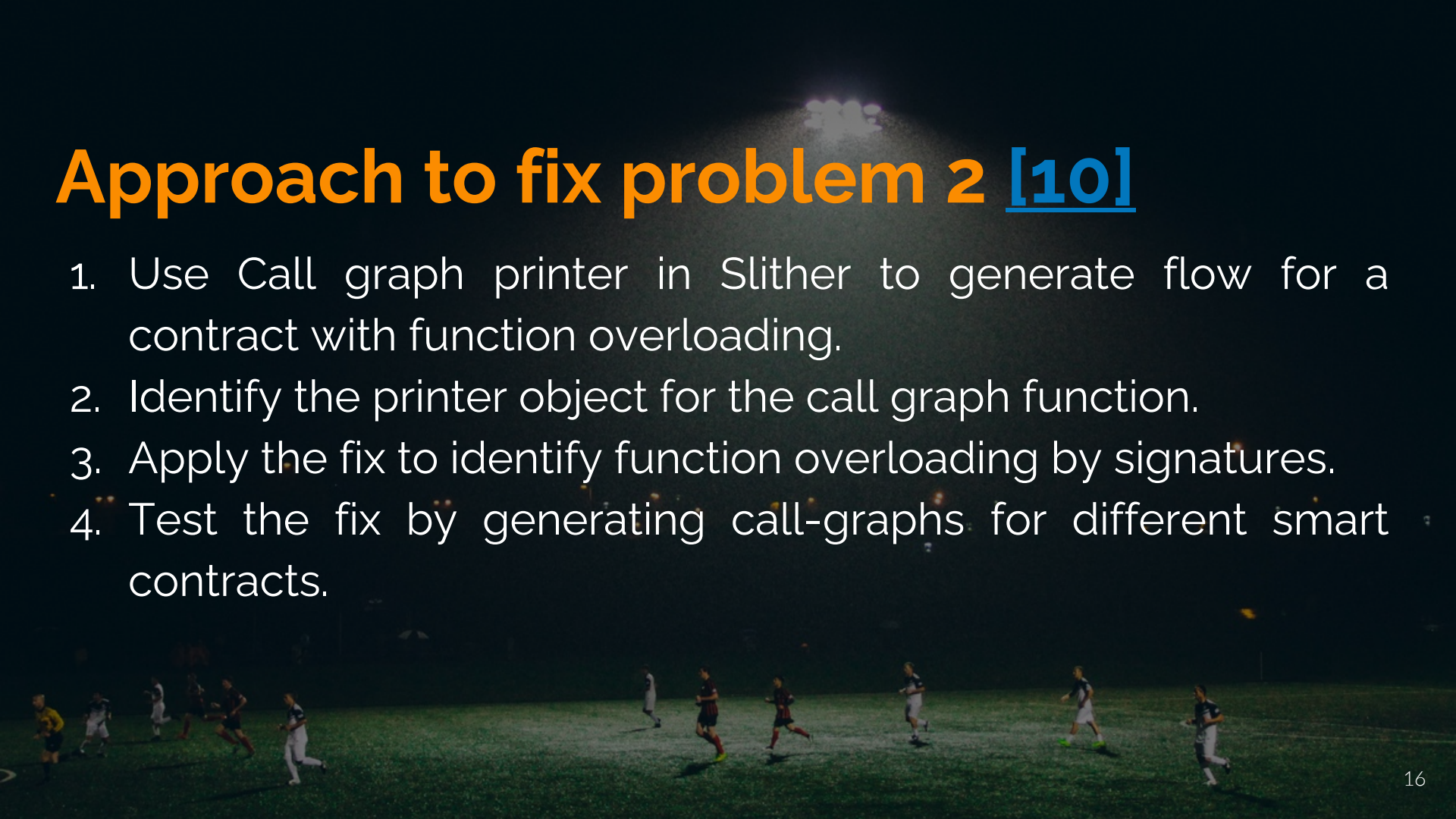
Issue 2 [10]:

Effects

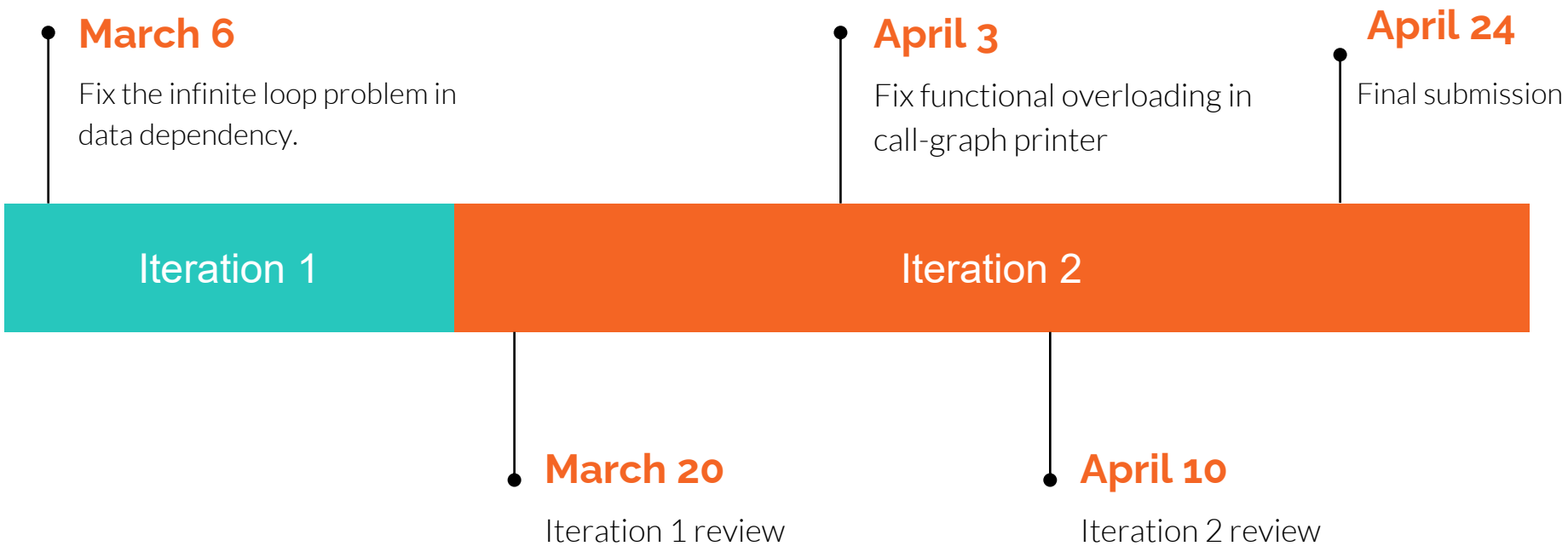
- Incorrect call-graph generation for a contract.
- Compatibility and effect on compile-time during compilation.

Approach to fix problem 2 [10]

1. Use Call graph printer in Slither to generate flow for a contract with function overloading.
2. Identify the printer object for the call graph function.
3. Apply the fix to identify function overloading by signatures.
4. Test the fix by generating call-graphs for different smart contracts.



Expectations



References

- [1]https://www.sfox.com/blog/how-secure-is-ethereum/, accessed 2/12/23.
- [2]https://ieeexplore.ieee.org/document/8823898, accessed 2/12/23.
- [3]https://ieeexplore.ieee.org/document/8823898, accessed 2/12/23.
- [4]https://arxiv.org/abs/1908.09878, accessed 2/12/23.
- [5]https://i0.wp.com/blog.trailofbits.com/wp-content/uploads/2019/05/overview.png?ssl=1 , accessed 2/12/23.

References

- [6]https://github.com/crytic/slither/issues/1127 , accessed 2/12/23.
- [7]https://pythontect.com/python-infinite-loop,accessed 2/12/23.
- [8]https://www.researchgate.net/publication/333700886_Slither_A_Static_Analysis_Framework_For_Smart_Contracts, accessed 2/12/23.
- [9]https://github.com/crytic/slither/issues/664, accessed 2/12/23.
- [10]https://bshastry.github.io/2018/11/05/Deconstructing-ToBs-Slither.html, accessed 2/12/23.

References

- [11]https://blog.trailofbits.com/2019/05/27/slither-the-leading-static-analyzer-for-smart-contracts/ , accessed 2/12/23.

Repository links:

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

THANK YOU