CSE 6324 Advance Topics in Software Engineering
Inception(Written Deliverable)
Slither: A Static Analysis Framework for Smart Contracts

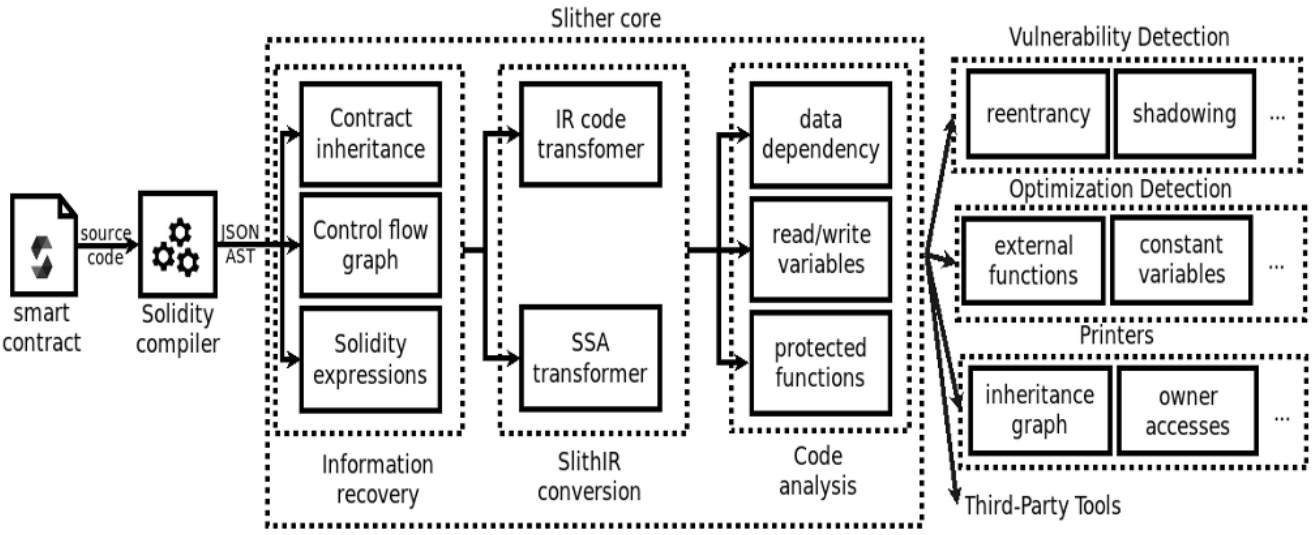
Team 6
Vaishnavi Khosla (1001906765)
Suresh Kavadi (1002040703)
Karthik Babu Vadloori (1002064678)
Danie Samanvitha Mellam (1002133148)

Table of Contents:

Abstract:

1. Slither, a framework for static analysis, was created for the examination of Ethereum smart contracts.
2. Intermediate Representation: It transforms Solidity contracts into SlithIR while keeping key details.
3. Analytical Capabilities: Slither provides dataflow and taint tracking analysis.
4. Use Cases
   - Vulnerability Detection: Automatically identifies contract vulnerabilities.
   - Code Optimization: Looks for ways to make the code more efficient.
   - Improved Understanding: Assists users in better comprehending smart contracts.
   - Code Review Assistance: Increases the effectiveness and dependability of code review.

Architecture:



Competitors:

Focused on Static Analysis: Slither scans Solidity code for flaws, whereas Hardhat, Brownie, and Truffle provide a complete development environment that includes contract compilation, testing, and deployment.

Advanced Detection: For typical Solidity contract vulnerabilities, including reentrancy hazards, unchecked calls, uninitialized state variables, and more, Slither provides a comprehensive set of detectors. When compared to many other tools, it excels at advanced and detailed detection.

Customizable Detectors: Users of Slither can create custom detectors to uncover problems unique to their smart contracts or project specifications. In some other programs, this level of flexibility is not as easily accessible.

Integration with Other Tools: Slither integrates without difficulty with other programs and processes. Static analysis improves development frameworks like Hardhat and Truffle, and security checks can be automated in CI/CD pipelines.

Multiple Output Formats: Slither offers a variety of output choices, including human-readable reports and support for JSON or CSV formats for simple automation and interaction with other applications.

Features:

Exceptional Value to Ethereum Developers:
By identifying flaws and boosting code security in Solidity, Slither offers crucial value to Ethereum developers and auditors, ensuring the dependability of blockchain applications.

Clearly Defined User Group:
Slither serves Ethereum programmers, auditors, and smart contract specialists, providing for the requirements of individuals and groups engaged in blockchain development who require reliable static analysis for smart contracts.

Core Features Identified and Analyzed:
With its vast detector library, adaptable detectors for project-specific difficulties, smooth integration with development workflows, and a variety of output formats, Slither excels.

Risk-Based Analysis:

Through the usage of its detectors, Slither enables users to identify and prioritize pressing problems. To help consumers better comprehend contract-related risks, it provides in-depth reports.

Risks:

1. **Evolving Ethereum Ecosystem (High Risk)**
    - Probability: High
    - Impact: Moderate

2. **Security Vulnerabilities in SlithIR(Moderate Risk)**
    - Probability: Moderate
    - Impact: Moderate
3. **Dependency on External Libraries(High Risk)**
    - Probability: Moderate
    - Impact: High

4. **Failure of Implementation(High Risk)**
    - Probability: Moderate
    - Impact: High

5. **Team Availability(Low Risk)**
    - Probability: Low
    - Impact: Low

Plan to Deal with Risks:

1. **Risk 1:** Keep up with Ethereum upgrades and quickly adapt Slither to changes. For insights, work with the Ethereum community.

2. **Risk 2:** Execute extensive testing and security assessments on the intermediate SlithIR representation. Create a strategy for addressing found vulnerabilities.

3. **Risk 3:** Update and review dependencies frequently. Keep a backup strategy in place in case crucial libraries stop being supported.

4. **Risk 4:** The risk can be decreased by testing and debugging the code for faults.

5. **Risk 5:** Maintain a backup plan in case a team member is unavailable.

References:

1. [Github: Slither]
    - https://github.com/crytic/slither
2. [TrailofBits Blog] “Slither: The Leading Static Analyzer for Smart Contracts”
    - https://blog.trailofbits.com/2019/05/27/slither-the-leading-static-analyzer-for-smart-contracts/
3. [Slither] “Slither: A static analysis framework for smart contracts”
    - By Josselin Feist, Gustavo Grieco, Alex Groce. Aug. 2019
    - In Proc. IEEE/ACM 2nd International Workshop on Emerging
    - Trends in Software Engineering for Blockchain (WETSEB)
    - https://arxiv.org/abs/1908.09878

Team 6 Github Repository:
https://github.com/Suresh-uta/ASE_CSE_6324