



THE UNIVERSITY OF TEXAS
AT ARLINGTON

Advanced Topics in Software Engineering

CSE-6324-004

Inception (Written Deliverable)

Sharma, Arnav -1002070507

Ashraf, Abdul Rafay - 1001918598

Kamble, Nikhil - 1001837272

Mohammed, Ishraq - 1002082521

I. Project Proposal:

Solify is a web based static analysis tool that takes solidity code as input to analyze. It employs Data flow analysis techniques for finding vulnerabilities in solidity. Some of the key features includes:

- Automated Optimization identification
- Automated Vulnerability detection
- Code comprehension
- Code review
- A graphical representation

II. Competitors:

Solhint: Solhint is a statistical analysis CLI tool that uses antlr4-based implementation of the Solidity parser that enables efficient parsing and validation performance. It helps developers to identify and fix code quality issues in their smart contracts.

Limitations:

- Limited Scope
- False Positives
- Limited Security analysis
- Limited Functionality for Testing
- Incapable of vulnerability Detection

SmartCheck: is a static analysis tool for Ethereum smart contracts that helps to identify potential security vulnerabilities in the Solidity code. It uses a set of predefined patterns to analyze the code and detect issues such as reentrancy, integer overflow and underflow, transaction-ordering dependence, and other potential vulnerabilities.

Limitations:

- Limited Customization
- Limited analysis capabilities
- No Graphical Representation

How our tool is different from competitor:

- Customizable analysis
- Graphical Representation
- Comprehensive Analysis
- Enhanced Reporting and Visualization
- Dependency Graph
- Ease of accessibility as it is a user-friendly web-based tool.

III. Features:

Solify will have following features:

- Web based user interface
- Ease of accessibility
- Code Editor
- Validation to standard input
- Analysis of syntactical errors
- List of patterns to analyze
- Graphical visualization [4]

• Delivery & Schedule:

- Build a web application based on the required framework for Solify
- First Iteration:
 - Support for uploading .sol files
 - Interface for the code editor for Solidity programs
- Second Iteration:
 - Validator for Standardization
 - Identifier for Syntactical errors
 - Changes based on feedback from Iteration 1
- Third Iteration:
 - Parser for solidity programs [3]
 - Support for Graphical Visualization
 - Changes based on feedback from Iteration 2
- Fourth Iteration:
 - Rigorous testing
 - Code Optimization
 - Changes based on feedback from Iteration 3

IV. **Risks:**

- The utilization of the tool may present a threat to security due to the necessity for users to upload Solidity program files. To lessen this vulnerability, the tool has an additional feature in the form of a text box where the user can input the program, decreasing the likelihood of security being compromised.
- Due to Solidity's rapid evolution with two major updates taking place annually [2], it is highly likely that adjustments will continuously need to be made to accommodate the changes in keywords and other elements that are added.
- A visualization tool does not perform in-depth security analysis [4]. Therefore, it only provides a basic overview of the code and cannot replace a full security audit.

V. **Customers/Users:**

- **Users:**
 - Community of Solidity
 - EVM based solidity Developers
 - Beginners who want to dig deeper into program analysis tools
- **Collaboration and Feedback:**
 - Dr Christoph Csallner
 - Mohammad Rifat Arefin
 - Team 10

VI. **References:**

1. [Ethereum Smart Contract Analysis Tools: A Systematic Review](#)
2. [Solidity Programming Language](#)
3. [Solidity parser](#)
4. [Solidity Visualizer](#)