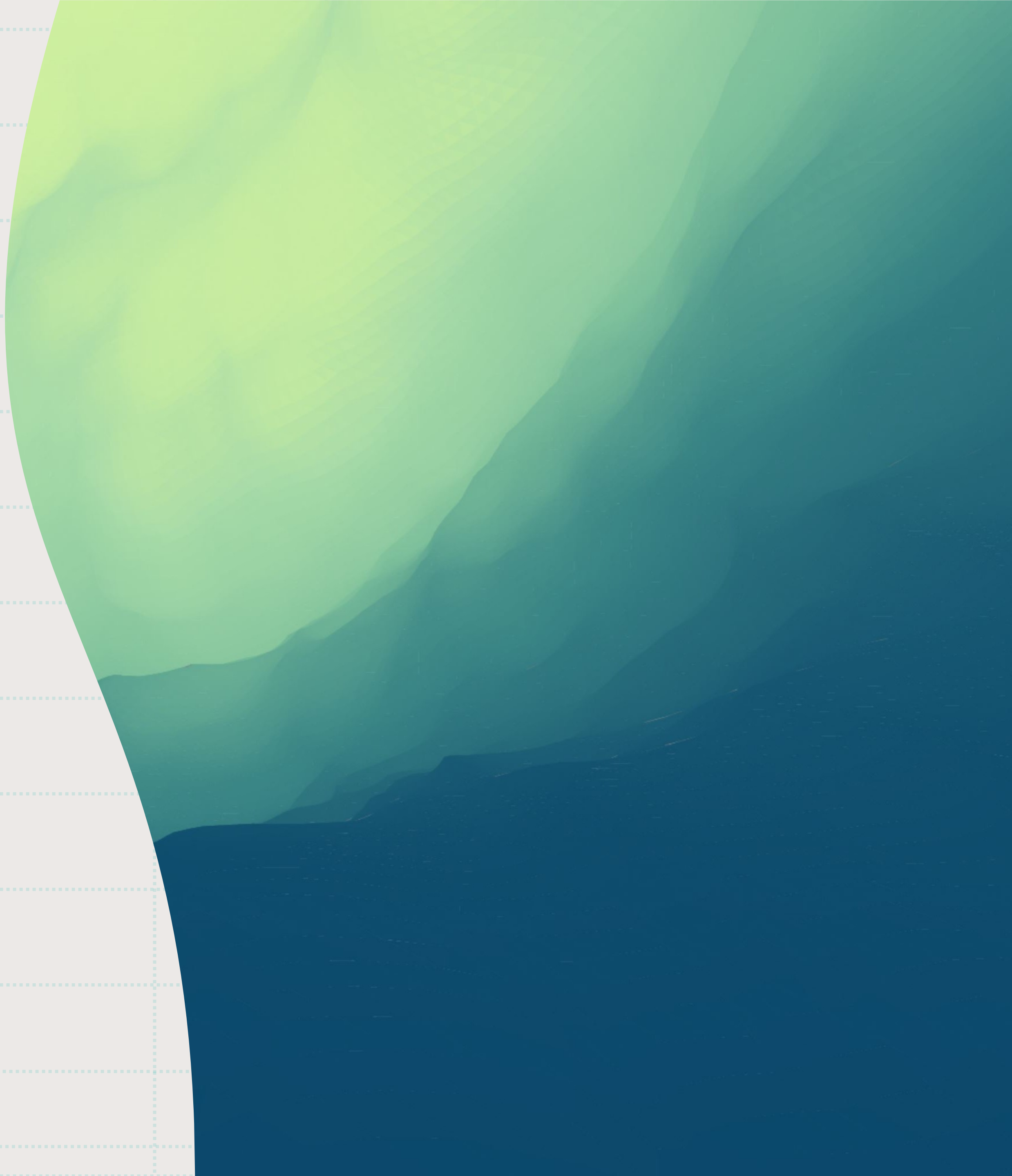
CSE 6324 Advanced Topics in Software Engineering

By – Team 2

1. Hema Sri Sesham(1001955263)
2. Moulin Chandrakant Jariwala (1001858654)
3. Parshva Urmish Shah(1001838879)
4. Vineeth Kumar Ananthula(1001953922)



Outline

Introduction

In this project we will be working on security analysis of the solidity smart contracts using conkas tool.

Conkas is a modular static analysis tool that use symbolic execution to end traces that lead to vulnerabilities and uses an intermediate representation (IR). The users can interact with Conkas via Command-Line.Interface (CLI) and the output will be the result of the analysis.

Architecture

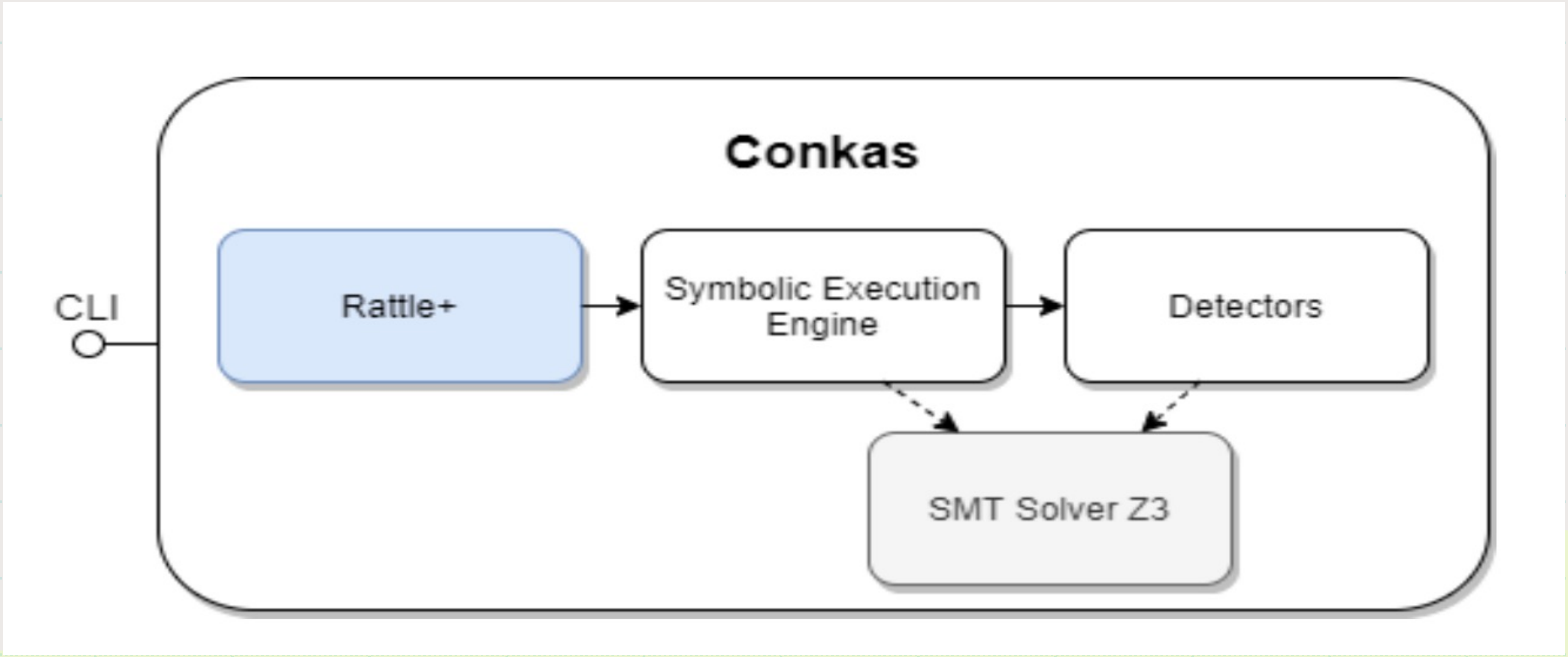


Figure 1: Architecture of Conkas tool. In Blue the modified module already existing. In grey the module already existing. [1]

Features of Conkas
[2]

- Arithmetic
- Reentrancy
- Time Manipulation
- Transaction Ordering Dependence
- Unchecked Low-Level Calls

Comparison to other tools

Several tools for smart contracts have been developed to address vulnerabilities detection. Other prominent tools that compete with this Conkas include Slither, Oyente, Osiris, Manticore, Mythril, and Securify, Smartcheck.

Using the SmartBugs framework, Conkas' performance was examined compared to that of other tools. When the true positive rate of all tools was calculated, Conkas had the highest percentage.

Target Users/Customers of Conkas

The main users of Conkas are the web3 developers and smart contract auditors. Auditing a Smart Contract is about extensively looking through the code to avoid any data breaches or bugs.

This is required because once this code is launched, it cannot be modified, which might result in massive security concerns, data loss, monetary loss and so on.

Hence, it is essential to use an analysis tool to detect and eliminate all the vulnerabilities in a contract before it is deployed.

What do we plan to do?

- We plan to detect the vulnerabilities regarding the **costly loops** and **tx.origin**.
- Costly loops and gas limit refers to -an attacker including infinite loops within an array to exhaust the gas limit mimicking DoS attack and freeing the transaction.
- Attacker can hide a withdraw function within the tx.origin variable to create a phishable contract and invoke a fallback function to steal all the funds.

Project Github Link

- https://github.com/shahparshva72/CSE6324-Team2-Project

References

- References:
- [1] Veloso, Nuno. Conkas: A Modular and Static Analysis Tool for Ethereum Bytecode - ULisboa. Jan. 2021, https://fenix.tecnico.ulisboa.pt/downloadFile/1689244997262417/94080-Nuno-Veloso_resumo.pdf.
-

[2] Polak, Kamil. "Hacking Solidity: Contracts Using Tx.origin for Authorization Are Vulnerable to Phishing." *HackerNoon*, 16 Jan. 2022, https://hackernoon.com/hacking-solidity-contracts-using-txorigin-for-authorization-are-vulnerable-to-phishing
-

[3] Sebastian Sinclair. "Blockchain Protocol Thorchain Suffers $8M Hack" Coindesk, 14 Sep. 2021,
- https://www.coindesk.com/markets/2021/07/23/blockchain-protocol-thorchain-suffers-8m-hack/