Project Inception

CSE 6324

Christoph Csallner

University of Texas at Arlington (UTA)

# Project Title: Conkas 2.0 - Static Analysis Tool

**Team - 6**

Apoorva Siri Mattewada - 1002026609
Harshith Kannalli Sachidananda Murthy -1001949874
Manisha Sohanlal Jain - 1001869817
Vinayak Neemkar - 1002022438
Vignaan Erram - 1002032121

# Project Vision

- Certain static analysis tools have low coverage to work with while dealing with smart contracts.
- Our goal is to increase the modules present in such tools, in order to help them find new vulnerabilities and add functionality to them.
- One such tool that exists is Conkas, which has 5 known vulnerabilities in total. [1]
- Our goal is to add additional modules to the existing Conkas tool.

# Features and Workflow

There are 5 categories of vulnerabilities that Conkas deal with:

1. Reentrancy
2. Arithmetic
3. Unchecked Low-Level Calls
4. Front-Running
5. Time Manipulation


[3]

We would try to extend conkas to cover the following vulnerabilities : [2]

- Short Address Attacks
- Bad Randomness

# Features & Workflow

- Iteration 1 Objective:

  To generate a vulnerability function for the existing code. Add a functionality to capture additional vulnerabilities in Conkas.

- Iteration 2 Objective:

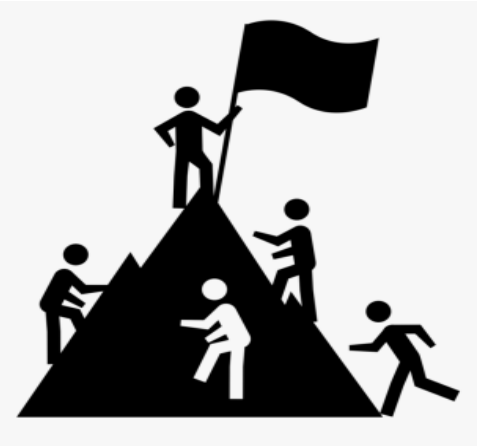
  Further working on adding new vulnerabilities to the same modules and testing it.

- Iteration 3 Objective:

  Work on existing vulnerabilities which are not found by the existing modules. Finally, test all the modules and run it on the tool successfully.

# Competitors

The Smartbugs framework was used to check vulnerabilities on few static analysis tools. In comparison to other tools, here's where Conkas stands.



[5]

| No. | Tool | Avg. Execution Time | Total Execution Time |
|---|---|---|---|
| 1 | Conkas | 0:00:32 | 1:14:37 |
| 2 | HoneyBadger | 0:01:12 | 2:49:03 |
| 3 | Maian | 0:03:47 | 8:52:25 |
| 4 | Manticore | 0:12:53 | 1 day, 6:15:28 |
| 5 | Mythril | 0:00:58 | 2:16:21 |

Table: Average execution time of each tool [1]

# Risks

- Some contracts contain vulnerabilities on their own, however, when these contracts are compiled, those vulnerabilities are gone because the vulnerability is in dead code. [1]
- Updating our tool regularly to counter changes in the contract as smart contracts evolve through time is tough.

[6]

- People from different development backgrounds come together to build the tool, this leads to language barriers, and working with such contracts without experience might pose financial risks as well.
- Dealing with false positive vulnerabilities can be tough as they come as a part of any static analysis tool.

# Customers and Users

- All smart contract developers in the business who need their contracts assessed for a variety of vulnerabilities can use this tool.
- Local entrepreneurs who deal in Ether and manage their own smart contracts.
- All professors and students who desire to do study on this type of tool are welcome to utilize it.
- People that want to modify it and add new features can adopt it.

# References


[1]https://www.semanticscholar.org/paper/Conkas%3A-A-Modular-and-Static-Analysis-Tool-for-Veloso/425e474177885f9ac9e57d44e8e2386d13f9c87d

[2] https://dasp.co/#item-7

[3]https://www.istockphoto.com/illustrations/clip-art-of-workflow

[4]https://www.pngitem.com/middle/TRhRTm_competition-competitive-advantage-business-clip-art-competition-clipart/

[5]https://www.wrike.com/blog/what-is-risk-identification-project-management/#What-is-risk-management

# Acknowledgements


- Dr. Christoph Csallner
- Shovon Pereira
- Mohammad Rafiq Arefin

# THANK YOU