# Improvement

## Cryptocurrency Development

Develop our own cryptocurrency following ERC-20 standard. I defined an interface at the very beginning and then implement a contract named F42Token which represent our group and has a symbol F42. Basically it obtained fundamental functions of a smart contract also being deployable [1].

## **Check Vulnerability**

After finishing rudiment of F42, I started to figure out what vulnerability it might has to further improve it. Suddenly the tool [2] provided in lecture came to my mind, so that I gave a try at once. The result is shown below as Figure 1 that did not provide sufficient information.



Figure 1: The result of detecting Solidity code vulnerability.

## **Detection Tool Construction**

We were told the website is rough and we are required to build another on our own. Then I started to build up a vulnerability detection tool by go through reference [3] one by one. Just as Figure 2 shown below, smart contracts may contain those vulnerabilities. Detection tools are various for instance, Slither, Mythril, Echidna, Manticore, ContractFuzzer [4], whereas Slither attracts me most. Slither simply detect vulnerabilities simultaneously [5] and what I need to do is extract critical information from analysis and draw a interactive website for us to use.

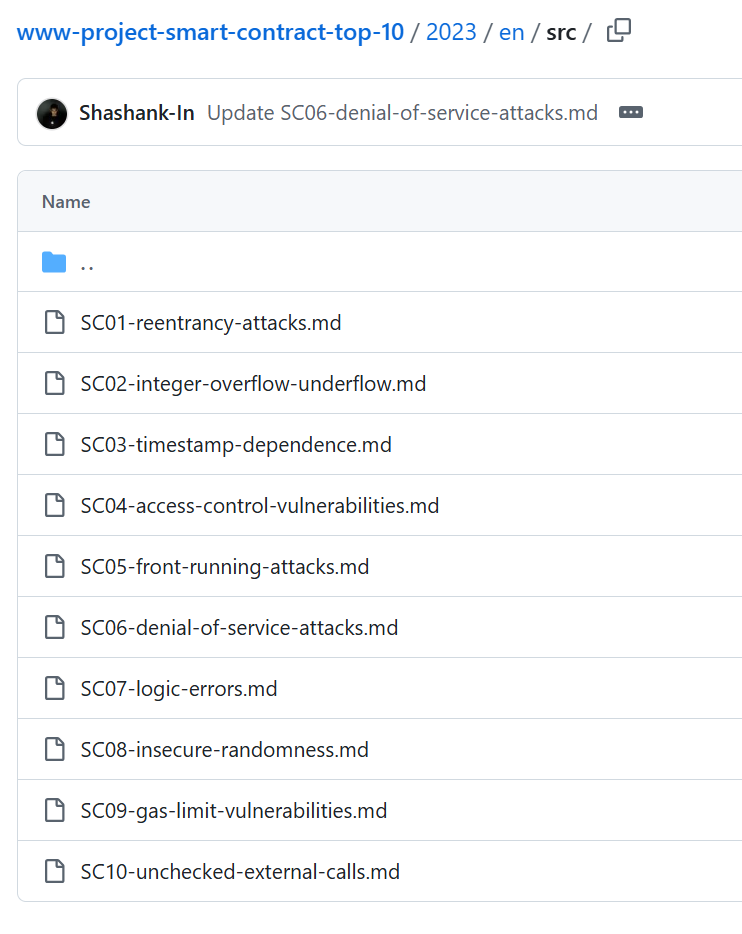


Figure 2: 10 possible Solidity code vulnerabilities.

## **Tool Website Deployment**

I chose Python Flask and HTML to form the website with HTML to display the content and Flask do calculation [6]. Then I detect F42 and found one vulnerability called reentrancy.

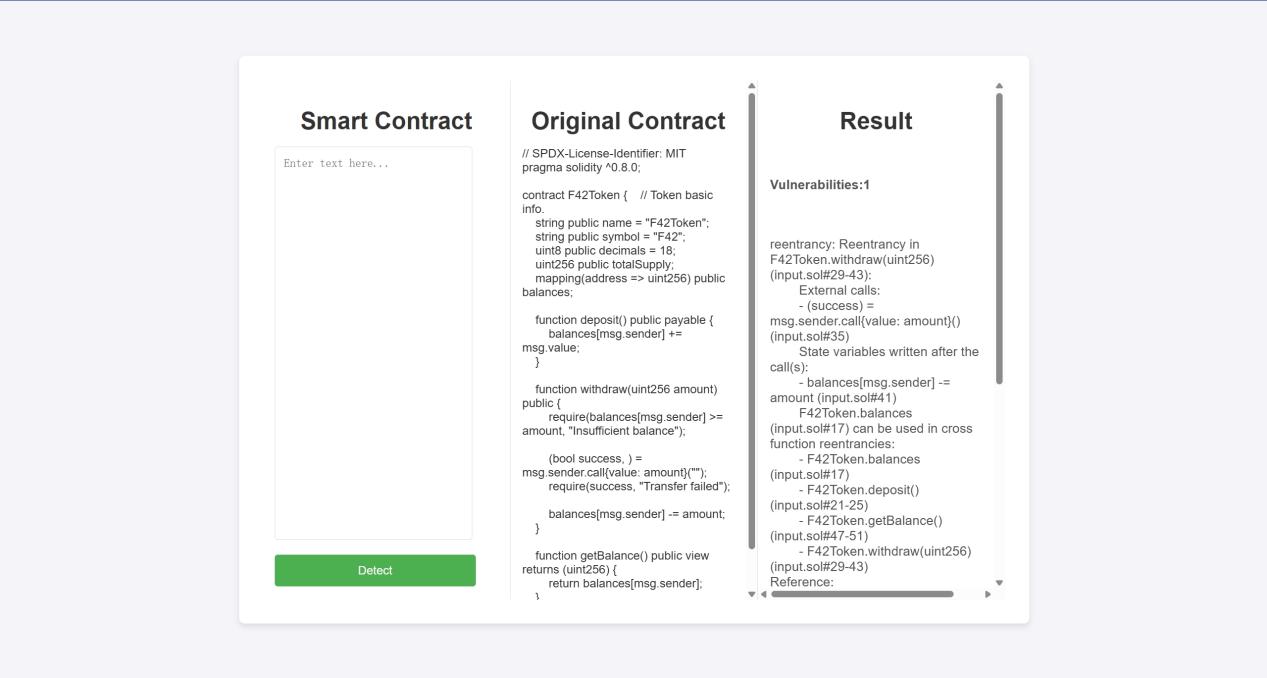


Figure 3: 1 vulnerabilities detected.

## **Solution**

I learned the reasons of reentrancy and reproduce it after deployment. Then modified F42 to reduce appearance of reentrancy attack. The key point is to manipulate balance before trading coins. Eventually vulnerabilities detection result reached 0 vulnerability as shown in Figure 4.

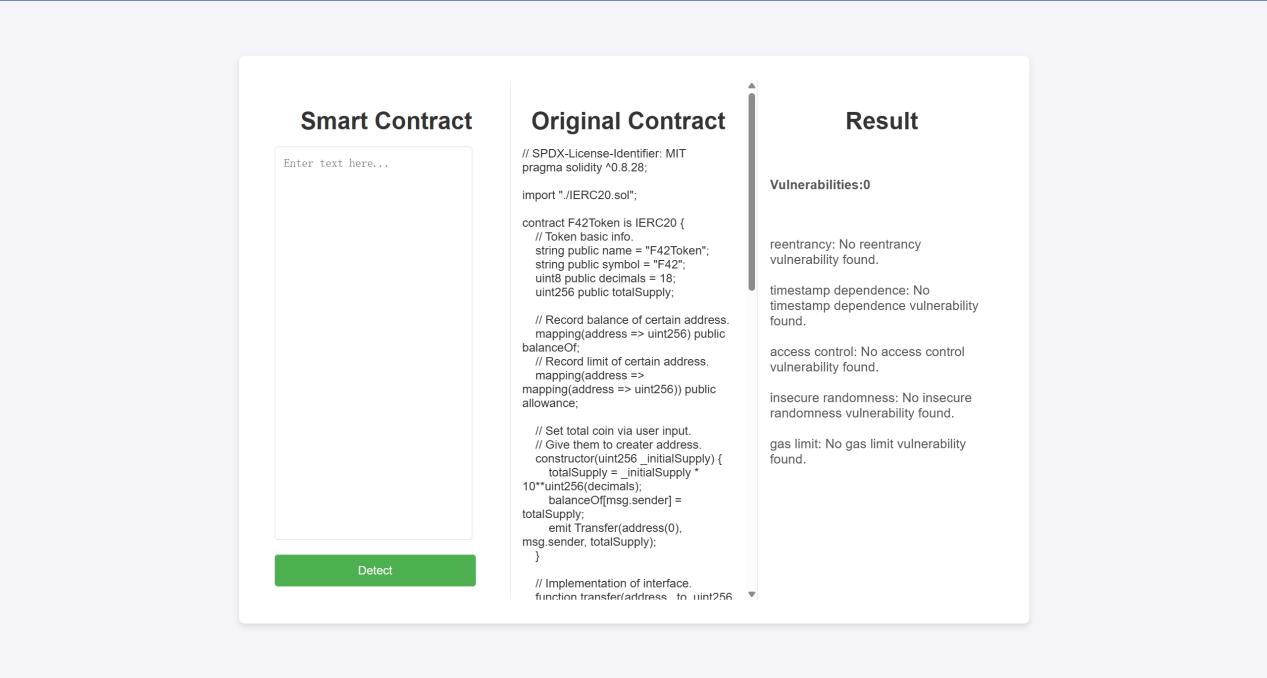


Figure 4: 0 vulnerabilities detected.

## **Conclusion**

All files are uploaded to GitHub and website is available.

## **reference**

1. [f42/work/2\_LSH\_Contract/F42 at main · RockyRori/f42](https://github.com/RockyRori/f42/tree/main/work/2_LSH_Contract/F42)
2. <https://lms.ln.edu.hk/mod/resource/view.php?id=739976>
3. <https://github.com/OWASP/www-project-smart-contract-top-10/tree/main/2023/en/src>
4. [A guide to smart contract security tools | ethereum.org](https://ethereum.org/en/developers/tutorials/guide-to-smart-contract-security-tools/)
5. [How to use Slither to find smart contract bugs | ethereum.org](https://ethereum.org/en/developers/tutorials/how-to-use-slither-to-find-smart-contract-bugs/)
6. [f42/work/2\_LSH\_Code/blockchain at main · RockyRori/f42](https://github.com/RockyRori/f42/tree/main/work/2_LSH_Code/blockchain)