SMART CONTRACTS SECURITY

OPERATING SYSTEMS AND CONCURRENCY #5CCS2OSC

Contributors of GROUP 46: Andrei-Bogdan Balcau, Iris Simionas, Sandra Popa, Livia-Oana Neagu, Zihao You, Sebastian Mesca

WHAT IS SMART CONTRACTS SECURITY?

- Smart Contracts Security is an emerging research area that deals with ***security*** issues arising from the execution of ***smart contracts*** in a blockchain system.

- Security breaches that surface on the main-net can cause massive financial losses or reputational damage.

- These SC security problems may be solved with the help of encryption.

- Ethereum also has a resource called GAS, that blocks the execution of smart contracts.

SMART CONTRACTS

Definition*: *Smart contracts are digital contracts that automatically process transactions when each of the encoded terms of the agreement is met by the ***transacting parties***.

Benefits:

1. Direct(no needed intermediaries)
2. Cost efficient
3. Time efficient
4. More secure
5. Extra-fraud resistant

Composability: They are public on ***Ethereum*** and can be thought of as open APIs.

Limitations: Smart contracts alone cannot get information about "real-world" events because they cannot send HTTPs requests.

ETHEREUM

1. PROGRAMMING LANGUAGE - SOLIDITY

- Object oriented.
- Influenced by C++, Python and JavaScript, target the Ethereum Virtual Machine(EVM).
- Statically typed, supports inheritance, libraries and complex user-defined types.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

2. PLATFORM - FRONT RUNNING

Displacement Scenario.

Insertion Scenario.

Suppression Scenario.

3. DENIAL-OF-SERVICE (DOS)

```
function distribute() public {
 require(msg.sender == owner);
 //only owner

 for(uint i=0; i < investors.length; i++)
 {
  transferToken(investors[i],
investorTokens[i]);

  //transfers "amount" of tokens to the
address
  }
}
```

Vulnerability:

Looping through externally manipulated mappings or arrays

Owner Operations

Progressing State Based on External Calls

Preventative Operations

4. REENTRANCY

Vulnerability

Preventative Operations

REFERENCES

- Docs.soliditylang.org. 2021. *Solidity — Solidity 0.8.4 documentation*. [online] Available at: <https://docs.soliditylang.org/en/develop/>
- Docs.soliditylang.org. 2021. *Introduction to Smart Contracts — Solidity 0.8.4 documentation*. [online] Available at: <https://docs.soliditylang.org/en/develop/introduction-to-smart-contracts.html#simple-smart-contract>
- Medium. 2021. *Front-running Attacks on Blockchain*. [online] Available at: <https://medium.com/codechain/front-running-attacks-on-blockchain-1f5ba28cd42b>
- Docs.soliditylang.org. 2021. *Solidity — Solidity 0.8.3 documentation*. [online] Available at: <https://docs.soliditylang.org/en/v0.8.3/>
- Sans.org. 2021. *Blockchain & Smart Contract Security | SANS SEC554*. [online] Available at: <https://www.sans.org/cyber-security-courses/blockchain-smart-contract-security/>.
- Research Paper 2021. [online] Available at: <https://www.researchgate.net/publication/336735093_Blockchain_Smart_Contracts_Formalization_Approaches_and_Challenges_to_Address_Vulnerabilities>
- Antonopoulos, A. and Wood, G., n.d. *Mastering Ethereum*.