Contract Audit
SymX

Smart Contract
Security Audit Report

Num: 07311750181151

Date: 2024-07-31

Welcome to SymX!

0x01 Summary Information

The SymX platform received this smart contract security audit application and audited the contract in Jul 2024.

It is necessary to declare that SymX only issues this report in respect of facts that have occurred or existed before the issuance of this report, and undertakes corresponding responsibilities for this. For the facts that occur or exist in the future, SymX is unable to judge the security status of its smart contract, and will not be responsible for it. The security audit analysis and other content made in this report are based on the documents and information provided to smart analysis team by the information provider as of the issuance of this report (referred to as "provided information"). SymX hypothesis: There is no missing, tampered, deleted or concealed information in the mentioned information. If the information that has been mentioned is missing, tampered with, deleted, concealed or reflected does not match the actual situation, SymX shall not be liable for any losses and adverse effects caused thereby.

Table 1 Contract audit information

Project	Description
Contract name	RealOldFuckMaker
Contract type	Ethereum contract
Code language	Solidity
Contract files	07311750181151.sol
Contract address	
Auditors	SymX team
Audit time	2024-07-31 17:50:20
Audit tool	SymX

Table 1 shows the relevant information of this contract audit in detail. The details and results of the contract security audit will be introduced in detail below.

0x02 Contract Audit Results

2.1 Vulnerability Distribution

The severity of vulnerabilities in this security audit is distributed according to the level of impact and confidence:

Table 2 Overview of contract audit vulnerability distribution

Vulnerability security distribution				
High	Medium	Low	Info	Opt
0	1	0	1	0

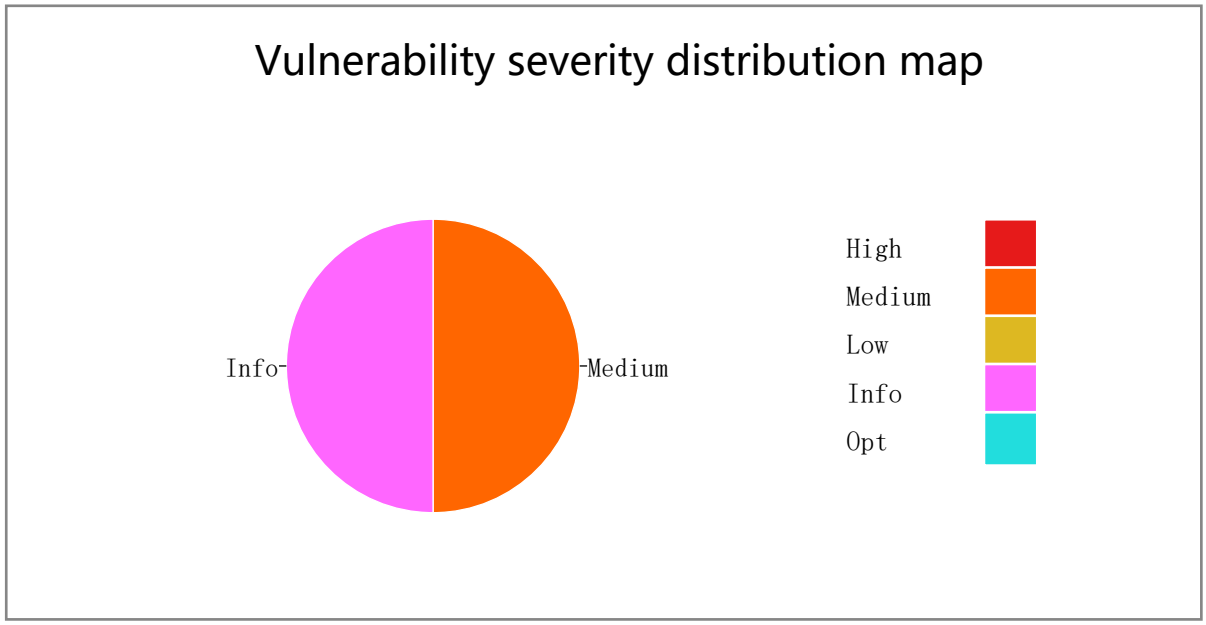

Figure 1 Vulnerability security distribution map

This security audit found 0 High-severity vulnerabilities, 1 Medium-severity vulnerabilities, 0 Low-severity vulnerabilities, 0 Optimization-severity vulnerabilities, and 1 places that need attention.

2.2 Audit Results

There are 18 test items in this security audit, and the test items are as follows (other unknown security vulnerabilities are not included in the scope of responsibility of this audit):

Table 3 Contract audit items

ID	Pattern	Description	Severity	Confidence	Status/Num
1	reentrancy-eth	Re-entry vulnerabilities (Ethereum theft)	High	probably	Pass
2	suicidal	Check if anyone can break the contract	High	exactly	Pass
3	controlled-delegatecall	The delegate address out of control	High	probably	Pass

4	arbitrary-send	Check if Ether can be sent to any address	High	probably	Pass
5	uninitialized-state	Check for uninitialized state variables	High	exactly	Pass
6	uninitialized-storage	Check for uninitialized storage variables	High	exactly	Pass
7	tod	Transaction sequence dependence for receivers/ethers	High	probably	Pass
8	incorrect-equality	Check the strict equality of danger	Medium	exactly	Pass
9	integer-overflow	Check for integer overflow	Medium	probably	Pass
10	unchecked-lowlevel	Check for uncensored low-level calls	Medium	probably	Medium:1
11	unchecked-send	Check unreviewed send	Medium	probably	Pass
12	tx-origin	Check the dangerous use of tx.origin	Medium	probably	Pass
13	timestamp	The dangerous use of block.timestamp	Low	probably	Pass
14	block-other-parameters	Hazardous use variables (block.number etc.)	Low	probably	Pass
15	low-level-calls	Check low-level calls	Info	exactly	Info:1
16	msgvalue-equals-zero	The judgment of msg.value and zero	Info	exactly	Pass
17	send-transfer	Check Transfe to replace Send	Opt	exactly	Pass
18	boolean-equal	Check comparison with boolean constant	Opt	exactly	Pass

0x03 Contract Code

3.1 Code

```
pragma solidity 0.4.24;

contract RealOldFuckMaker {
    address fuck = 0xc63e7b1DEcE63A77eD7E4Aeef5efb3b05C81438D;

    function makeOldFucks(uint32 number) {
        uint32 i;
        for (i = 0; i < number; i++) {
            //   UNCHECKED_LL_CALLS
            fuck.call(bytes4(sha3("giveBlockReward()")));
        }
    }
}
```

3.2 Contract CFG

The 1-th contract RealOldFuckMaker

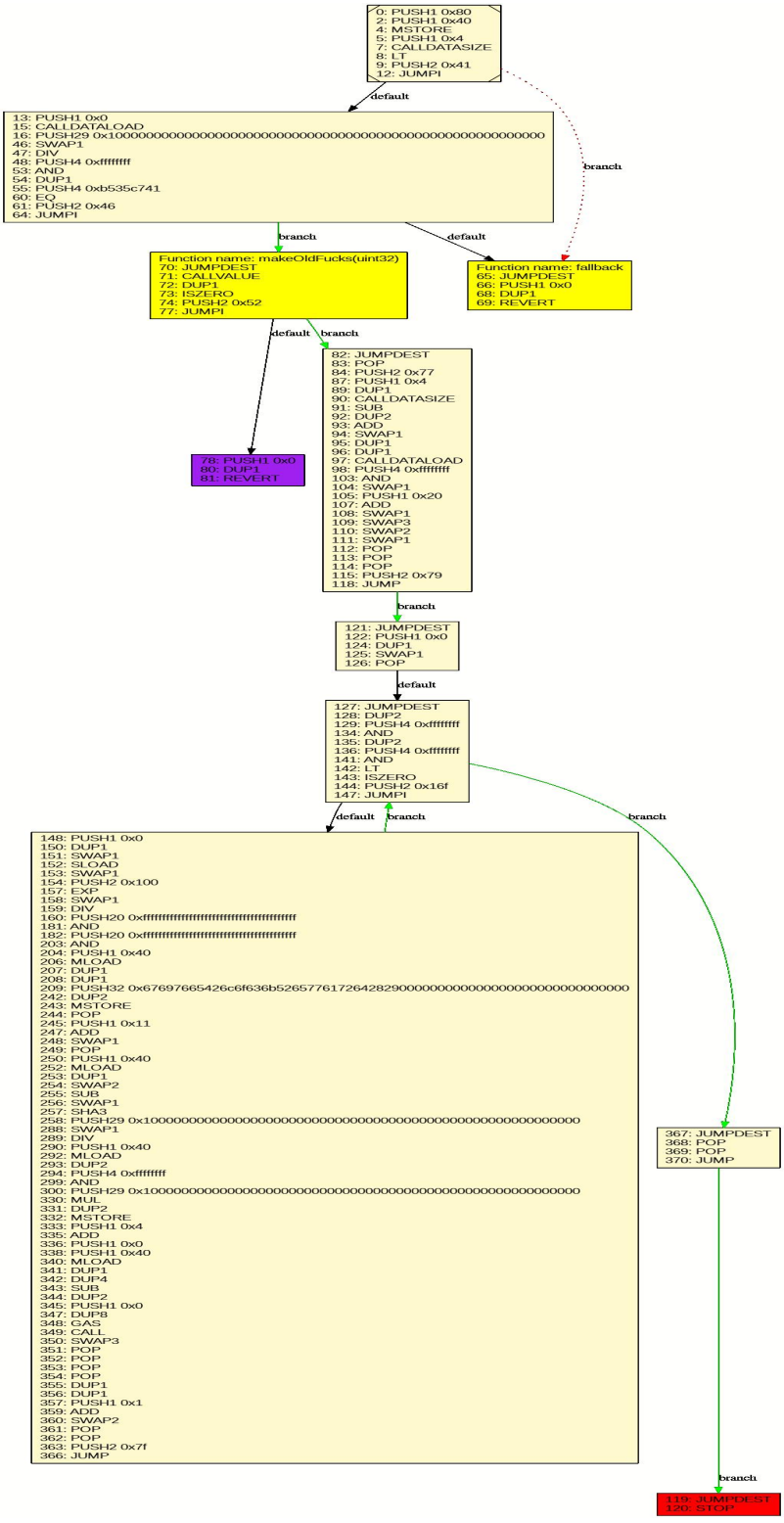

Figure 2 The CFG of contract RealOldFuckMaker.

0x04 Contract Audit Details

4.1 unchecked-lowlevel

Vulnerability description

The low-level call to the external contract failed, and the return value was not judged. When sending ether at the same time, please check the return value and handle the error.

Audit results: 【Medium:1】

For this pattern, the specific problems in the contract are as follows:

The 1-th problem is located at pc 0x15e

The defective code locates at line [10], and it is shown as follows:

```
10  fuck.call(bytes4(sha3("giveBlockReward()")))
```

Security advice

Make sure to check or record the return value of low-level calls.

4.2 low-level-calls

Vulnerability description

Label low-level methods such as call, delegatecall, and callcode, because these methods are easily exploited by attackers.

Audit results: 【Info:1】

For this pattern, the specific problems in the contract are as follows:

The 1-th problem is located at pc 0x15d

The defective code locates at line [10], and it is shown as follows:

```
10  fuck.call(bytes4(sha3("giveBlockReward()")))
```

Security advice

Avoid low-level calls. Check whether the call is successful. If the call is to sign a contract, please check whether the code exists.

Thanks for using the SymX contract audit platform!