GO CLC

CUBO LODGE CLUB

Abstract

In this report, we consider the security of GO Cubo Lodge Club, who engaged ForsetiDev team on 9 December 2017 to perform smart contracts audit of GO Token. The contract was audited and few minor bugs were found. These issues were corrected in 13 December. A second on was conducted on the corrected version, commit version 5375dba9eb0bad10892557abe59d70bd39821911 and no additional issues were found.

Analysis technique

We used several publicly available automated Solidity analysis tools, as well as proceed manual analysis. All the issues found by tools were manually checked (rejected or confirmed). Contracts were manually analyzed, their logic was checked and compared with the one described in the whitepaper.

Bugs classification

CRITICAL - problems leading to stealing funds from any of the participants, or making them inaccessible by anyone

SEVERE - problems that can stop, freeze or break the internal logic of the contract

WARNING - non-critical problems that cannot break the contract, but contract code does not match declared in WhitePaper logic

Notes - any other findings .

Automated Analysis

Oyente

Timestamp Dependency

CommonCrowdsale : line 357

Timestamp Dependency : True

GOTokenCrowdsale : line 357

Timestamp Dependency:True

Securify

Transaction Reordering

Transactions May Affect Ether Receiver

Matched lines: L.501

Transactions May Affects Ether Amount

Matched lines: L.501

All the issues found by tools were manually checked (rejected or confirmed). Cases, when these issues lead to actual bugs or vulnerabilities, are described in the next section.

Findings

No issues were found in the second audit

Recommendations

Undesirable loops

CommonCrowdsale.sol, line 431 :

```
function payExtraTokens(uint count)
```

Loops are undesirable and quite dangerous in solidity, we recommend avoid them where it possible.In this case, we recommend modifying this function in a way, that investor should initiate payExtraTokens by himself

CommonCrowdsale.sol, line 412 :

```
function end()
```

In this case its possible to declare variable “foo” and add it to function

```
uint256 foo;
………..
function addMilestone(uint periodInDays, uint discount) public onlyOwner {
 milestones.push(Milestone(periodInDays, discount));
 foo+= periodInDays;
}
```

and increment foo+=periodInDays
to modify end() function this way

```
function end() public constant returns(uint) {
 uint last = start+foo;
 return last;
}
```