Резюме

В этом отчёте, мы изучаем безопасность смарт контрактов проекта Cubo Lodge Club,обратившихся к нам для проведения аудита безопасности смарт контрактов 9 декабря 2017. Контракт был проаудирован , и несколько незначительных багов было найдено. Эти баги были исправлены 13 декабря . Второй аудит был проведен над исправленной версией смарт контрактов , версия коммита `5375dba9eb0bad10892557abe59d70bd39821911` от 13 декабря 2017 года. В результате аудита дополнительных багов не было найдено

Методика анализа

Код контракта просматривается вручную на наличие известных уязвимостей, ошибок в логике, соответствие WhitePaper. Также были использованы инструменты автоматического анализа кода. Все найденные инструментами ошибки были протестированы вручную, в результате чего, либо добавлены в отчет , либо отклонены

Классификация уязвимостей

КРИТИЧНЫЕ - возможность кражи эфира/токенов или их блокировки без возможности восстановления доступа или иной потери эфира/токенов.

СЕРЬЕЗНЫЕ - возможность нарушений работы контракта, при которых для восстановления его корректной работы необходима модификация состояния контракта вручную или его полная замена.

ПРЕДУПРЕЖДЕНИЯ - возможность нарушения запланированной логики контракта или не соответствие объявленной логики в WhitePaper

ЗАМЕЧАНИЯ - все остальные замечания.

Автоматический анализ

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

Все найденные уязвимости были проверены вручную и те из них, которые ведут к багам, отражены в отчёте.

Найденные уязвимости

Не было найдено уязвимостей в результате второго аудита

Рекомендации

Нежелательные циклы

CommonCrowdsale.sol, line 431 :

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

Циклы не желательны в солидити, там, где возможно, лучше их избегать. В данном случае, мы рекомендуем переделать эту функцию, таким образом, чтобы инвестор сам инициировал

CommonCrowdsale.sol, line 412 :

```
function end()
```

В данном случае можно завести переменную “foo” и добавить в функцию

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

делать foo+=periodInDays

и тогда в функция end() примет вид

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