KeySecurity

GamblingDAO Security Review

Duration: Octomber 8, 2024 - Octomber 11, 2024

Date: November 29, 2024

Conducted by: **KeySecurity**
gkrastenov, Lead Security Researcher

Table of Contents

1 About KeySecurity

KeySecurity is a new, innovative Web3 security company that hires top-talented security researchers for your project. We have conducted over 30 security reviews for various projects, collectively holding over $300,000,000 in TVL. For security audit inquiries, you can reach out to us on Twitter/X or Telegram @gkrastenov or check our previous work `here`.

2 About GamblingDAO

GamblingDAO is a GameFi project where participants in their game can select a cube by paying a fee. If their cube is gold, silver, or bronze, they can win a reward; otherwise, their fee is added to the total reward pool.

3 Disclaimer

Audits are a time, resource, and expertise bound effort where trained experts evaluate smart contracts using a combination of automated and manual techniques to identify as many vulnerabilities as possible. Audits can show the presence of vulnerabilities **but not their absence**.

4 Risk classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

4.1 Impact

- **High** - leads to a significant loss of assets in the protocol or significantly harms a group of users.
- **Medium** - only a small amount of funds can be lost or a functionality of the protocol is affected.
- **Low** - any kind of unexpected behaviour that’s not so critical.

4.2 Likelihood

- **High** - direct attack vector; the cost is relatively low to the amount of funds that can be lost.
- **Medium** - only conditionally incentivized attack vector, but still relatively likely.
- **Low** - too many or too unlikely assumptions; provides little or no incentive.

4.3 Actions required by severity level

- **Critical** - client **must** fix the issue.
- **High** - client **must** fix the issue.
- **Medium** - client **should** fix the issue.
- **Low** - client **could** fix the issue.

5 Executive summary

Overview

Project Name	GamblingDAO
Repository	https://github.com/GamblingDAO/1mc-contracts
Commit hash	27dd1e145b935d5c0a0835a90147633467c4c330
Review Commit hash	ccf24a0b742f3c9a89d8b3a392a6c37aaf3dc996
Documentation	https://github.com/GamblingDAO/1mc-contracts
Methods	Manual review

Scope

OneMillionCubes.sol

Timeline

Octomber 8, 2024	Audit kick-off
Octomber 11, 2024	Preliminary report
November 29, 2024	Mitigation review

Issues Found

Severity	**Count**
High	0
Medium	1
Low	3
Information	3
Total	**7**

6 Findings

6.1 Medium

6.1.1 Insufficient funds in the contract can block withdraw mechanism

Severity: *Medium*

Context: Global

Description: In some scenarios, it is possible for the game reward mechanism to be blocked, resulting in either users or the owner losing funds. All of the scenarios below are possible because users randomly pick cubes, and the user rewards come from the provided fees during the selection process.

For the following examples, the following parameters will be used:

`fee` = 0.1; `minDiscElemNumb` = 100; `singleGoldReward` = 1 ether; `numOfGoldRewards` = 3;

`singleSilverReward` = 0.5 ether; `numOfSilverRewards` = 5`singleBronzeReward` = 0.1 ether;

`numOfBronzeRewards` = 10

Case 1:

In a scenario where we have 600 selected cubes and all of the gold, silver, and bronze cubes are selected, not all of the rewards can be withdrawn. In this case, the contract balance will be 6 ether, but the total sum of rewards is 6.5 ether. If the gold medal is selected last, all future participants may stop selecting cubes because all of the winnable cubes have already been chosen.

The winner of the gold cube will not be able to withdraw their reward. Alternatively, if the owner of the contract decides to urgently withdraw all funds and pay an additional 0.5 ether to cover their reward, the owner will lose 0.5 ether.

Case 2:

In the following scenario, if we have 120 cubes selected, including 1 gold and 1 silver. Two users are allowed to claim their rewards, but again, the smart contract does not have enough balance to pay the rewards of all users.

If the silver cube is selected first and the silver reward is claimed, and then the gold cube is selected, the user who picks the gold cube must wait for 30 selections of other cubes by other users, which may not happen. This could lead to the user who chose the gold cube not being able to withdraw their rewards, resulting in the loss of their provided fees. Alternatively, the owner of the contract may need to urgently withdraw all funds and pay an additional 0.3 ether out of pocket to cover all rewards, total 1 ether.

Recommendation: Change the game reward mechanism to ensure that all possible scenarios prevent users from being able to misuse funds.

Resolution and Client comment: Acknowledged.

6.2 Low

6.2.1 The msg.sender address can be used as the referrer

Severity: *Low*

Context: OneMillionCubes.sol#L187

Description: Every time a user selects a cube, they can provide an address for referral. The referral system can be used to incentivize users to recommend this game by offering bonuses outside of the game. Every user can gain an advantage by using their own address for referral (`_referrer == msg.sender`).

Recommendation: Make the following changes:

```
-if (_referrer != address(0)) {
+if (_referrer != address(0) && _referrer != msg.sender) {
            referralCounts[_referrer]++;
            emit CubeSelectedWithReferral(
                msg.sender,
                _x,
                _y,
                msg.value,
                _referrer
            );
  } else {
          emit CubeSelected(msg.sender, _x, _y, msg.value);
  }
```

Resolution and Client comment: Resolved. Fixed at #ccf24a0b742f3c9a89d8b3a392a6c37aaf3dc996 commit.

6.2.2 Users can select cubes with coordinates that are out of range

Severity: *Low*

Context: OneMillionCubes.sol#L170-L171

Description: During cube selection, the user picks random x and y coordinates, which determine their cube. In the selection process, it is never checked whether the x and y coordinates are out of range.

For example, if the total number of cubes is 10,000, the maximum allowed value for x and y would be 100. However, if the user chooses for x or y values greater than 100, their coordinates will be out of range, allowing them to pick invalid cubes and lose their fee.

Recommendation: Check if the provided values for `_x` and `_y` are not out of range in the `selectCube` function.

Resolution and Client comment: Acknowledged.

6.2.3 minDiscElemNumb can not be reached

Severity: *Low*

Context: OneMillionCubes.sol#L214

Description: if the `minDiscElemNumb` is not reached all participant will lose their fees.

One of the important requirements for a user to withdraw their reward is that the total selections must be greater than `minDiscElemNumb` (`totalSelections < gameParameters.minDiscElemNumb = false`). If not enough users participate in selecting the cubes (for example, if only 75 cubes are selected and no more users participate), their fees will be stuck in the contract with no chance to withdraw their provided fees if they see that the game is frozen and does not have much interest from other users.

Recommendation: Allow users to withdraw their fees if the `minDiscElemNumb` is not reached after a certain period of time.

Resolution and Client comment: Acknowledged.

6.3 Information

6.3.1 Redundant error

Severity: *Information*

Context: OneMillionCubes.sol#L108

Description: In the `OneMillionCubes` contract, the error `InvalidCoordinates` is never used.

Recommendation: Remove the redundant error.

Resolution and Client comment: Resolved. Fixed at #ccf24a0b742f3c9a89d8b3a392a6c37aaf3dc996 commit.

6.3.2 Users can start selecting cubes before the registration phase

Severity: *Information*

Context: OneMillionCubes.sol#L208

Description: Users can start selecting cubes before registering all the gold, silver, and bronze cubes.

The initial logic of the `OneMillionCubes` contract is for the owner to register all the gold, silver, and bronze cubes, after which users can start selecting cubes.

Recommendation: Allow users to select cubes only after registering all the gold, silver and bronze cubes.

Resolution and Client comment: Acknowledged.

6.3.3 Use msg.sender instead of owner()

Severity: *Information*

Context: OneMillionCubes.sol#L270

Description: When a function is only accessible by the owner due to the presence of the `onlyOwner` modifier, there is no need to call an internal function to obtain the owner's address. In this case, `msg.sender` is equal to `owner()`.

Recommendation: Make the following changes:

```
- (bool sent, ) = payable(owner()).call{value: balance}("");
+ (bool sent, ) = payable(msg.sender).call{value: balance}("");

- emit EmergencyWithdrawal(owner(), balance);
+ emit EmergencyWithdrawal(msg.sender, balance);
```

Resolution and Client comment: Resolved. Fixed at #ccf24a0b742f3c9a89d8b3a392a6c37aaf3dc996 commit.