CERTIK

Security Assessment

DODO Mining

Mar 31st, 2021

Summary

This report has been prepared for DODO Mining smart contracts, to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Dynamic Analysis, Static Analysis, and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in 9 findings that ranged from miner to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases given they are currently missing in the repository;
- Provide more comments per each function for readability, especially contracts are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	DODO Mining
Description	Decentralized Mining Project
Platform	Ethereum, BSC
Language	Solidity
Codebase	https://github.com/DODOEX/contractV2/tree/c84b224399b17423642a9606aa9f29295a95b177
Commits	c84b224399b17423642a9606aa9f29295a95b177

Audit Summary

Delivery Date	Mar 31, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

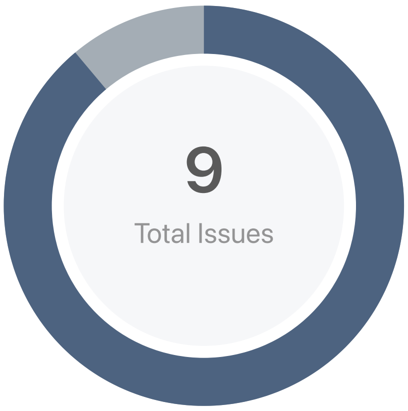
Vulnerability Summary

Total Issues	9
● Critical	0
● Major	0
● Minor	1
● Informational	8
● Discussion	0

Audit Scope

ID	file	SHA256 Checksum
BME	BaseMine.sol	e4b4bc2b61472169c17ea9c2daa7c82628f1714a9e37b7df80f231abc4e156c3
ERC	ERC20Mine.sol	10c19f58becbbf3cdef9355609b37bcd2061478eddd64ae7958b086a24ca9c5b
RVT	RewardVault.sol	887dd53af05a242140380590a276aff4794706a4ce8bfe7bf22af7ecabaca95c
DOD	vDODOMine.sol	abc28cfe605df825d5ec1e94a2a33f38d0aa0e295066aa452b59a2f9ded70f05

Findings



🟥 **Critical**	**0** (0.00%)
🟧 **Major**	**0** (0.00%)
⬜ **Minor**	**1** (11.11%)
🟦 **Informational**	**8** (88.89%)
🟩 **Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
BME-1	Wrong Warning Message	Logical Issue	● Informational	✓ Resolved
BME-2	Wrong Parameter of Require	Logical Issue	● Minor	✓ Resolved
BME-3	Code Optimization for Function BaseMine.removeRewardToken()	Gas Optimization	● Informational	✓ Resolved
BME-4	Proper Usage of “public” And “external” Type	Gas Optimization	● Informational	✓ Resolved
BME-5	Discussion For Function BaseMine.claimReward() and IRewardVault.reward()	Logical Issue	● Informational	✓ Resolved
RVT-1	Discussion For Function BaseMine.claimReward() and IRewardVault.reward()	Logical Issue	● Informational	✓ Resolved
DOD-1	Discussion For function vDODOMine.syncBalance()	Logical Issue	● Informational	✓ Resolved
DOD-2	Missing Emit Events	Control Flow	● Informational	✓ Resolved
DOD-3	Missing Transfer Process	Logical Issue	● Informational	✓ Resolved

BME-1 | Wrong Warning Message

Category	Severity	Location	Status
Logical Issue	● Informational	BaseMine.sol: 77	✓ Resolved

Description

Wrong Warning message of check in function `BaseMine.getRewardTokenById()`:

```
require(i<rewardTokenInfos.length, "DODOMineV2: REWARD_ID_FOUND");
```

Recommendation

Consider correcting warning message : "DODOMineV2: REWARD_ID_NOT_FOUND".

Alleviation

The team heeded our advice and corrected the warning message. Code change was applied in commit : 327b56ea15e8d922d8e973b51ff73058642fa947.

BME-2 | Wrong Parameter of Require

Category	Severity	Location	Status
Logical Issue	● Minor	BaseMine.sol: 89	✓ Resolved

Description

When `rewardToken` not found in function `BaseMine.getIdByRewardToken()`, it should return a warning message.But according to the logic,it won't return anything.

Recommendation

Consider modifying like this :

```
require(false, "DODOMineV2: TOKEN_NOT_FOUND");
```

Alleviation

The team heeded our advice and corrected the require parameter. Code change was applied in commit : 327b56ea15e8d922d8e973b51ff73058642fa947.

BME-3 | Code Optimization for Function BaseMine.removeRewardToken()

Category	Severity	Location	Status
Gas Optimization	● Informational	BaseMine.sol: 146	✓ Resolved

Description

If i equal to len-1,we don't need to execute code `rewardTokenInfos[i] = rewardTokenInfos[len - 1];`.

Recommendation

Consider adding a check like this :

```
 if(i != len-1) {
      rewardTokenInfos[i] = rewardTokenInfos[len - 1];
}
rewardTokenInfos.pop();
emit RemoveRewardToken(rewardToken);
break;
```

Alleviation

The team heeded our advice and added the if judgement. Code change was applied in commit : 327b56ea15e8d922d8e973b51ff73058642fa947.

BME-4 | Proper Usage of “public” And “external” Type

Category	Severity	Location	Status
Gas Optimization	● Informational	BaseMine.sol: 64~66, 76~80, 105~110	✓ Resolved

Description

`public` functions that are never called by the contract could be declared `external`.

Recommendation

Consider using the external attribute for functions never called from the contract : `BaseMine.getPendingRewardByToken()` , `BaseMine.getRewardTokenById()` , `BaseMine.claimAllRewards()`.

Alleviation

The team heeded our advice and changed the function type. Code change was applied in commit : 327b56ea15e8d922d8e973b51ff73058642fa947.

BME-5 | Discussion For Function BaseMine.claimReward() and IRewardVault.reward()

Category	Severity	Location	Status
Logical Issue	● Informational	BaseMine.sol: 100	✓ Resolved

Description

We notice that the function `RewardVault.reward()` can only be executed by the owner. But in function `BaseMine.claimReward()` the reward is transfered to `msg.sender`. So how can other users claim their own rewards. Is there any other implementation of `IRewardVault.reward()`.

Alleviation

(DODO response) the owner of the function `RewardVault.reward()` is the mining contract.Users who participated in mining can execute function `BaseMine.claimReward()` to transfer reward to their own address.

RVT-1 | Discussion For Function BaseMine.claimReward() and IRewardVault.reward()

Category	Severity	Location	Status
Logical Issue	● Informational	RewardVault.sol: 28	✓ Resolved

Description

We notice that the function `RewardVault.reward()` can only be executed by the owner. But in function `BaseMine.claimReward()` the reward is transfered to `msg.sender`. So how can other users claim their own rewards. Is there any other implementation of `IRewardVault.reward()`.

Alleviation

(DODO response) the owner of the function `RewardVault.reward()` is the mining contract.Users who participated in mining can execute function `BaseMine.claimReward()` to transfer reward to their own address.

DOD-1 | Discussion For function vDODOMine.syncBalance()

Category	Severity	Location	Status
Logical Issue	● Informational	vDODOMine.sol: 58~69	✓ Resolved

Description

Please introduce when will the function `vDODOMine.syncBalance()` be executed and what will happen.

Alleviation

(DODO response) they have mining business both on ethereum chain and bsc chain. And vDODO is on ethereum chain. DODO hope users on bsc chain can also participate in vDODO mining.So they need to synchronize datas from ethereum to bsc.

DOD-2 | Missing Emit Events

Category	Severity	Location	Status
Control Flow	● Informational	vDODOMine.sol: 58~69	✓ Resolved

Description

Function `vDODOMine.syncBalance()` should be able to emit events as notifications to customers because it change the status of sensitive variables.

Recommendation

Consider adding an emit after change the status of variables as below :

```
    function syncBalance(address[] calldata userList) external {
        for (uint256 i = 0; i < userList.length; ++i) {
            address user = userList[i];
            uint256 curBalance = balanceOf(user);
            uint256 vDODOBalance = IERC20(_vDODO_TOKEN_).balanceOf(user);
            if (curBalance > vDODOBalance) {
                _updateAllReward(user);
                _totalSupply = _totalSupply.add(vDODOBalance).sub(curBalance);
                _balances[user] = vDODOBalance;
            }
        }
        emit ...;
    }
```

Alleviation

The team heeded our advice and added the event. Code change was applied in commit : 38180be331cf9d22769660a433001c59a1d31020

DOD-3 | Missing Transfer Process

Category	Severity	Location	Status
Logical Issue	● Informational	vDODOMine.sol: 46, 55	✓ Resolved

Description

Contract `vDODOMine` missing transfer process in function `deposit()` and `withdraw()`.

Alleviation

(DODO response) the function `transfer()` is turned off in contract `vDODOToken` now.So they just record the balance in `vDODOMine`,and contact with contract `vDODOToken` through function `getLockedvDODO()`.

Appendix

Finding Categories

Gas Optimization

Gas Optimization findings refer to exhibits that do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Mathematical Operations

Mathematical Operation exhibits entail findings that relate to mishandling of math formulas, such as overflows, incorrect operations etc.

Logical Issue

Logical Issue findings are exhibits that detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Data Flow

Data Flow findings describe faults in the way data is handled at rest and in memory, such as the result of a struct assignment operation affecting an in-memory struct rather than an in storage one.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete .

Coding Style

Coding Style findings usually do not affect the generated byte-code and comment on how to make the codebase more legible and as a result easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

Magic Numbers

Magic Number findings refer to numeric literals that are expressed in the codebase in their raw format and should otherwise be specified as constant contract variables aiding in their legibility and maintainability.

Compiler Error

Compiler Error findings refer to an error in the structure of the code that renders it impossible to compile using the specified version of the project.

Disclaimer

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we’re able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.