Coinsult

Request your audit at coinsult.net

Advanced Manual

Smart Contract Audit

May 1, 2023

Audit requested by

Moox LP Stake Manager

0x6e23bc4d0c8607f1a3f809270df9654223decfe6

Audit Summary

Project Name	Moox
Website	moox.one
Blockchain	Binance Smart Chain
Smart Contract Language	Solidity
Contract Address	0x6e23bc4d0c8607f1a3f809270df9654223decfe6
Audit Method	Static Analysis, Manual Review
Start date of audit	May 1, 2023

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identified.

The audit of the Solidity codebase presented significant challenges due to its disorganized nature and lack of accompanying documentation. The convoluted structure of the code made it increasingly difficult to discern the intended functionality of individual components. Furthermore, the absence of any guiding documentation surrounding the functions made it even more arduous to review and assess the overall quality of the code. Despite these obstacles, the auditing process was diligently carried out to ensure the highest standards of security and stability.

Audit Scope

Source Code

Coinsult was comissioned by MooX to perform an audit based on the following code:

https://github.com/mooxtoken/moox-contract/tree/master/contracts/v2

Note that we only audited the code available to us on this URL at the time of the audit. If the URL is not from any block explorer (main net), it may be subject to change. Always check the contract address on this audit report and compare it to the token you are doing research for.

Audit Method

Coinsult's manual smart contract audit is an extensive methodical examination and analysis of the smart contract's code that is used to interact with the blockchain. This process is conducted to discover errors, issues and security vulnerabilities in the code in order to suggest improvements and ways to fix them.

Automated Vulnerability Check

Coinsult uses software that checks for common vulnerability issues within smart contracts. We use automated tools that scan the contract for security vulnerabilities such as integer-overflow, integer-underflow, out-of-gas-situations, unchecked transfers, etc.

Manual Code Review

Coinsult's manual code review involves a human looking at source code, line by line, to find vulnerabilities. Manual code review helps to clarify the context of coding decisions. Automated tools are faster but they cannot take the developer's intentions and general business logic into consideration.

Used Tools

- ✓ Slither: Solidity static analysis framework
- ✓ Remix: IDE Developer Tool
- ✓ CWE: Common Weakness Enumeration
- ✓ SWC: Smart Contract Weakness Classification and Test Cases
- ✓ DEX: Testnet Blockchains

Audit Results

Risk Classification

Coinsult uses certain vulnerability levels, these indicate how bad a certain issue is. The higher the risk, the more strictly it is recommended to correct the error before using the contract.

Vulnerability Level	Description
Informational	Does not compromise the functionality of the contract in any way
Low-Risk	Won't cause any problems, but can be adjusted for improvement
Medium-Risk	Will likely cause problems and it is recommended to adjust
High-Risk	Will definitely cause problems, this needs to be adjusted

Manual Code Review

In this audit report we will highlight the following issues:

Vulnerability Level	Total	Pending	Acknowledged	Resolved
Informational	0	0	0	0
Low-Risk	5	0	3	2
Medium-Risk	0	0	0	0
High-Risk	0	0	0	0

Error Code	Description	Severity
CNS-026	LpStakeManager.sol: The return value of an external transfer/transferFrom call is not checked	Logical Fixed

Issue

The return value of an external transfer/transferFrom call is not checked

Code

```
lpToken.transferFrom(address(msg.sender), address(this), _amount);

lpToken.transfer(address(msg.sender), _amount);

safeMooxTransfer(address(msg.sender), pending); - not fixed

lpToken.transfer(address(msg.sender), amountToTransfer);
```

Recommendation

Check the return value

Moox Comment

Fixed

Coinsult Comment

Also fix it for safeMooxTransfer

Error Code	Description	Severity
CNS-027	LpStakeManager.sol: Strict equality	Logical Acknowledged

Issue

Use of strict equalities that can be easily manipulated by an attacker.

Code

```
        if (lpTokenSupply == 0) {
            lastRewardDepositedMoox = totalDepositedMoox;
            return;
        }
```

Recommendation

Don't use strict equality to determine if an account has enough Ether or tokens.

Moox Comment

...

Error Code	Description	Severity
CNS-028	LpStakeManager.sol: No events emitted	Logical Acknowledged

Issue

The contract LpStakeManager was found to be missing these events on the function mooxDeposit() which would make it difficult or impossible to track these transactions off-chain

Code

```
function mooxDeposit(uint256 _amount) external nonReentrant {
    require(msg.sender == address(moox), 'StakeManager: Not allowed');

    totalDepositedMoox = totalDepositedMoox.add(_amount);
}
```

Recommendation

Emit an event.

Moox Comment

Acknowledged, won't fix.

Error Code	Description	Severity
CNS-029	LpStakeManager.sol: Outdated floating solidity version	Logical Acknowledged

Issue

Outdated floating versions were detected pragma solidity ^0.8.0;

Code

```
pragma solidity ^0.8.0;
```

Recommendation

Use a fixed recent version of silidity

Moox Comment

Acknowledged, won't fix.

Error Code	Description	Severity
CNS-030	LpStakeManager.sol: Gas optimization	Logical Fixed

Issue

State variables that are not updated following deployment should be declared immutable to save gas. Should be immutable.

Code

```
// The Moox token
IBEP20 public moox;

// The LP token for MOOX/BUSD
IPancakeswapPair public lpToken;
```

Recommendation

Make the constant immutable

Moox Comment

Fixed

Notes

Notes by Moox

No notes provided by the team.

Notes by Coinsult

These contracts have been audited based on GitHub code, this code is updateable.

Disclaimer

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identified.

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.

The audit of the Solidity codebase presented significant challenges due to its disorganized nature and lack of accompanying documentation. The convoluted structure of the code made it increasingly difficult to discern the intended functionality of individual components. Furthermore, the absence of any guiding documentation surrounding the functions made it even more arduous to review and assess the overall quality of the code. Despite these obstacles, the auditing process was diligently carried out to ensure the highest standards of security and stability.