CERTIK

Preliminary Comments

MOBOX-Farm

Apr 14th, 2021

Summary

This report has been prepared for MOBOX-Farm 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 findings that ranged from critical 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	MOBOX-Farm
Platform	BSC
Language	Solidity
Codebase	https://github.com/moboxio/NFTfarmer/tree/f3b6a2176ced7828867d366648063f3b672cd212
Commits	9443add71a3dadfefab237ac2b016c942d16e204

Audit Summary

Delivery Date	Apr 14, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

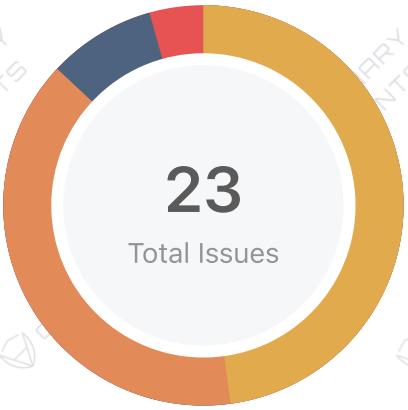
Vulnerability Summary

Total Issues	23
Critical	1
Major	9
Minor	11
Informational	2
Discussion	0

Audit Scope

ID	file	SHA256 Checksum
KTM	KeyToken.sol	0ca758d4ea6dfcc2570330f5f3e917fc7d5e57d2c48e574e0565006db9cc5e4d
LIC	LICENSE	662e1446749c9fcd09d44ad7dec8f6bd506d7a4b1a206cba2e42f56cb3e94174
MFM	MoboxFarm.sol	175b601762befc9dc4e228144ea1c14d1a52305a886e1ec057f5b67773c05048
MSP	MoboxStrategyP.sol	f4fdac6c357352990cfe144ac9417eb67597aba9e76b0ce64ccb3ee5f0cfda48
MSV	MoboxStrategyV.sol	5953f3c3ad07bc482fea4d32f13dd7eeadedbefc25efb48e37b5329d443413a1
MTM	MoboxToken.sol	40ada3a9edf02f55eb862482e85d56cafc998da8794822c4b739c0c56b65b0a1
MSM	MomoStaker.sol	569c40585b02e7daad1adf2300146f04b51a70357905428a3e5889578fae6ecc
REA	README.md	b1c6017652aaba25e46108d9f561366a164f6874758b191750e63e460f23f7a5
AMO	comm/Address.sol	f5aa76bbc397d9fa4fc1823b3a87551199afbc3070950d7dfbbccfa18d84a060
CMO	comm/Context.sol	cad53f9540d93519d2931c4616718e8fadd214b294947b585912418bc7b4035c
ERC	comm/ERC20.sol	bd75e3222674e1c647ee70c0ffcb2bb85093dcf7fff573d744ed139534cf7d20
IER	comm/IERC1155.sol	65df2a59fbd11dcf991b2433861efe4c18f76c3e1f61167d5f338e4c5d513513
IEC	comm/IERC20.sol	18ed31aa23dc283989d53c8e1b1bfd16fe0e0ae3b86e850307bf9d7ceea58b35
IEM	comm/IERC721.sol	356060fe385961d8163716736e2e8eba5f98754753c1b6066726fa5de9de08ee
OMO	comm/Ownable.sol	40c9c9493cb0d38b5b26b1abd762736d12b7320f4b2b0dc1ad0b2751432459fe
PMO	comm/Pausable.sol	73bfe964db22014edd7ff337ce2f68c13a90ab758fb5994378c5ebb40dba4588
RGM	comm/ReentrancyGuard.sol	00ed2d907800a389c8d6bb5fd01153ce01de4bee0237bcd8d79629061d6f05b9
SER	comm/SafeERC20.sol	05333159ad7c695c7188a839babc7a309ecbacf957c5512ff2192d15e3bf5aec
SMM	comm/SafeMath.sol	25e89aec7527d143b48a15a9483d9e334690fba0bc0818a94166cd30034d920b

Findings



Critical	1 (4.35%)
Major	9 (39.13%)
Minor	11 (47.83%)
Informational	2 (8.70%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
KTM-01	Missing zero address validation	Volatile Code	Minor	Pending
MFM-01	No return values	Logical Issue	Major	Pending
MFM-02	Contract sets array length with a user controlled value	Volatile Code	Minor	Pending
MFM-03	Missing Return Value Handling	Logical Issue	Minor	Pending
MFM-04	Dangerous usage of block.timestamp	Logical Issue	Minor	Pending
MFM-05	Missing Checks for Reentrancy	Logical Issue	Major	Pending
MSP-01	Integer Overflow	Mathematical Operations	Minor	Pending
MSP-02	Missing Checks for Reentrancy	Logical Issue	Major	Pending
MSP-03	Missing zero address validation	Logical Issue	Minor	Pending
MSP-04	Missing slippage protection	Logical Issue	Minor	Pending
MSP-05	Wrong withdraw amount	Logical Issue	Major	Pending
MSP-06	No return values	Logical Issue	Major	Pending
MSP-07	Unimpletation constructor function	Volatile Code	Informational	Pending
MSV-01	Integer Overflow	Mathematical Operations	Minor	Pending

ID	Title	Category	Severity	Status
MSV-02	Missing slippage protection	Logical Issue	Minor	Pending
MSV-03	Compile error	Compiler Error	Critical	Pending
MSV-04	Missing Checks for Reentrancy	Logical Issue	Major	Pending
MSV-05	Missing access control	Logical Issue	Major	Pending
MSV-06	No return values	Logical Issue	Major	Pending
MSV-07	Ignored return values	Logical Issue	Major	Pending
MSV-08	Leverage risk	Logical Issue	Informational	Pending
MTM-01	Ignored return value	Logical Issue	Minor	Pending
PMO-01	Compiler Error	Compiler Error	Minor	Pending

KTM-01 | Missing zero address validation

Category	Severity	Location	Status
Volatile Code	Minor	KeyToken.sol: 36	Pending

Description

lacks a zero-check on : - moboxFarm = farm_ (KeyToken.sol#36)

Recommendation

Adding check that farm_ is not zero.

MFM-01 | No return values

Category	Severity	Location	Status
Logical Issue	Major	MoboxFarm.sol: 251	Pending

Description

Functions define with return values but no actual value is returned.

Recommendation

Confirm return values to be returned.

MFM-02 | Contract sets array length with a user controlled value

Category	Severity	Location	Status
Volatile Code	Minor	MoboxFarm.sol: 132~140	Pending

Description

MoboxFarm (MoboxFarm.sol#35-510) contract sets array length with a user-controlled value: - poolInfoArray.push(PoolInfo(wantToken_,SafeMathExt.safe32(allocPoint_),uint64(block.timestamp),0,0,strategy (MoboxFarm.sol#132-139)

Recommendation

Do not allow array lengths to be set directly set; instead, opt to add values as needed. Otherwise, thoroughly review the contract to ensure a user-controlled variable cannot reach an array length assignment.

MFM-03 | Missing Return Value Handling

Category	Severity	Location	Status
Logical Issue	Minor	MoboxFarm.sol: 110~111	Pending

Description

Approve is not a void-returning function per IERC20 interface. Ignoring the return value might cause some unexpected exception, especially if the callee function doesn't revert automatically when failing.

Recommendation

We recommend checking the return value before continuing processing.

MFM-04 | Dangerous usage of block.timestamp

Category	Severity	Location	Status
Logical Issue	Minor	MoboxFarm.sol: 126, 170, 181, 206, 229, 234, 297, 309	Pending

Description

block.timestamp can be manipulated by miners.

Recommendation

Avoid relying on block.timestamp.

MFM-05 | Missing Checks for Reentrancy

Category	Severity	Location	Status
Logical Issue	Major	MoboxFarm.sol: 461~462, 178, 123, 450, 107, 148, 201, 328	Pending

Description

Functions have state updates or event emits after external calls are vulnerable to reentrancy attack.

Recommendation

We recommend applying OpenZeppelin ReentrancyGuard library - nonReentrant modifier for the aforementioned functions to prevent reentrancy attack.

MSP-01 | Integer Overflow

Category	Severity	Location	Status
Mathematical Operations	Minor	MoboxStrategyP.sol: 266, 242, 252, 313	Pending

Description

Although integer overflows would not happen if some variables such as now are within regular ranges, SafeMath is still highly recommended for mathematical operations, if gas costs are not considered as a most significant factor in implementations, to prevent exceptions.

Recommendation

We recommend applying SafeMath.add at the aforementioned line

MSP-02 | Missing Checks for Reentrancy

Category	Severity	Location	Status
Logical Issue	Major	MoboxStrategyP.sol: 266	Pending

Description

Function harvest() have state updates or event emits after external calls and thus are vulnerable to reentrancy attack.

Recommendation

Applying nonReentrant modifier.

MSP-03 | Missing zero address validation

Category	Severity	Location	Status
Logical Issue	Minor	MoboxStrategyP.sol: 319	Pending

Description

lacks a zero-check on :

- strategist = strategist_ (MoboxStrategyP.sol#319)

Recommendation

Adding check that strategist_ is not zero address.

MSP-04 | Missing slippage protection

Category	Severity	Location	Status
Logical Issue	Minor	MoboxStrategyP.sol: 239, 249, 310	Pending

Description

Missing slippage protection in all swapExactTokensForTokensSupportingFeeOnTransferTokens() functions.

Recommendation

Limit max slippage.

MSP-05 | Wrong withdraw amount

Category	Severity	Location	Status
Logical Issue	Major	MoboxStrategyP.sol: 165~171	Pending

Description

If lpBalance > amount_, then wantAmount still set to lpBalance.

165 uint256 lpBalance = IERC20(wantToken).balanceOf(address(this)); 166 if (lpBalance < amount_) { 169 } 170 171 uint256 wantAmount = lpBalance;

Recommendation

Confirm this is intended, or set wantAmount to amount_.

MSP-06 | No return values

Category	Severity	Location	Status
Logical Issue	Major	MoboxStrategyP.sol: 198~199, 129	Pending

Description

Functions define with return values but no actual value is returned.

Recommendation

Confirm return values to be returned.

MSP-07 | Unimpletation constructor function

Category	Severity	Location	Status
Volatile Code	Informational	MoboxStrategyP.sol: 84~86	Pending

Description

`constructor() public` Unimpletation constructor function

Recommendation

Impletation constructor function,should delete if not needed

MSV-01 | Integer Overflow

Category	Severity	Location	Status
Mathematical Operations	Minor	MoboxStrategyV.sol: 407, 447	Pending

Description

Although integer overflows would not happen if some variables such as now are within regular ranges, SafeMath is still highly recommended for mathematical operations, if gas costs are not considered as a most significant factor in implementations, to prevent exceptions.

Recommendation

We recommend applying SafeMath.add at the aforementioned line

MSV-02 | Missing slippage protection

Category	Severity	Location	Status
Logical Issue	Minor	MoboxStrategyV.sol: 404, 444	Pending

Description

Missing slippage protection in all swapExactTokensForTokensSupportingFeeOnTransferTokens() functions.

Recommendation

Limit max slippage.

MSV-03 | Compile error

Category	Severity	Location	Status
Compiler Error	Critical	MoboxStrategyV.sol: 450~451	Pending

Description

Error: Expected '(' but got identifier --> MoboxStrategyV.sol:452:14:

Recommendation

Fix the compile error by moving the "}" to an new line.

MSV-04 | Missing Checks for Reentrancy

Category	Severity	Location	Status
Logical Issue	Major	MoboxStrategyV.sol: 329~330, 282, 248, 248, 316, 321	Pending

Description

Many functions have state updates or event emits after external calls and thus are vulnerable to reentrancy attack.

Recommendation

We recommend applying OpenZeppelin ReentrancyGuard library - nonReentrant modifier for the aforementioned functions to prevent reentrancy attack.We recommend applying OpenZeppelin ReentrancyGuard library - nonReentrant modifier for the aforementioned functions to prevent reentrancy attack.

MSV-05 | Missing access control

Category	Severity	Location	Status
Logical Issue	Major	MoboxStrategyV.sol: 103~104	Pending

Description

It's unsafe that init function can be called by anyone.

Recommendation

Add onlyOwner modifier.

MSV-06 | No return values

Category	Severity	Location	Status
Logical Issue	Major	MoboxStrategyV.sol: 161, 360	Pending

Description

Functions define with return values but no actual value is returned.

Recommendation

Confirm return values to be returned.

MSV-07 | Ignored return values

Category	Severity	Location	Status
Logical Issue	Major	MoboxStrategyV.sol: 133~149	Pending

Description

Ignored return values may cause unexpected risks. For example, if repayBorrow() failed then leverage can't decreased while no place aware this failure.

Recommendation

Check every function return values.

MSV-08 | Leverage risk

Category	Severity	Location	Status
Logical Issue	Informational	MoboxStrategyV.sol: 268	Pending

Description

This strategy uses leverage, which may introduce potential risk. E.g If the Strategist doesn't deleverage in time, the system may enter liquidity crisis in an extreme market.

Recommendation

Don't use leverage.

MTM-01 | Ignored return value

Category	Severity	Location	Status
Logical Issue	Minor	MoboxToken.sol: 107~115	Pending

Description

MoboxFarm.setMoMoMinter(address) (MoboxFarm.sol#107-115) ignores return value by IERC20(keyToken).approve(momoMinter,0) (MoboxFarm.sol#110)

Function `setMoMoMinter` calls `IERC20().approve()`, but does not handle the result.

Recommendation

Ensure that all the return values of the function calls are used.

PMO-01 | Compiler Error

Category	Severity	Location	Status
Compiler Error	Minor	comm/Pausable.sol: 32	Pending

Description

Met below compiler error:

Error: No visibility specified. Did you intend to add "public"?

Recommendation

Check function types

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

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without CertiK's prior written consent.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

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.