CERTIK

Security Assessment

Monkey NFT Game

Apr 3rd, 2021

Summary

This report has been prepared for Monkey NFT Game 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 20 findings that ranged from major to informational. We recommend to address these findings as potential improvements that can benefit the long run as both smart contracts would lock a significant amount of tokens for a significant amount of time. We have done rounds of communications over the general understanding and the team has resolved the questions promptly.

Overall the source code is well written with security practices. The business logic is straightforward and implemented accordingly. Yet we suggest a few recommendations that could better serve the project from the security perspective:

1. Enhance general coding practices for better structures of source codes;
2. Add enough unit tests to cover the possible use cases given they are currently missing in the repository;
3. Provide more comments per each function for readability, especially contracts are verified in public.

Overview

Project Summary

Project Name	Monkey NFT Game
Description	NFT
Platform	OKExChain
Language	Solidity
Codebase	monkeynftgame/monkeynft
Commits	8702c0ff1491bb0a7813da9231c72db734d5538f

Audit Summary

Delivery Date	Apr 03, 2021
Audit Methodology	Manual Review
Key Components	

Vulnerability Summary

Total Issues	20
● Critical	0
● Major	1
● Minor	8
● Informational	11
● Discussion	0

Audit Scope

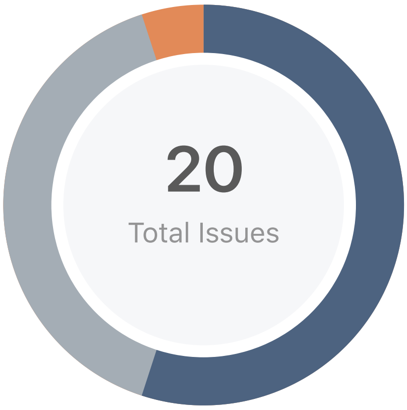
ID	file	SHA256 Checksum
BTN	monkeyNFT0325/BoxesToken.sol	ede52c035f128f05ff54e6ac68a694c22383a4f556d43260e323edf91571603c
BVN	monkeyNFT0325/BoxesV1.sol	4edb8383ac99df53ce481dc0f15f9dbdaa991e318e9c3ff94dd93916d412d6c3
FVN	monkeyNFT0325/FeedV1.sol	1b571829c2f976ebc2db64be8047706168812aad3daeaa0cf90a356b9f523d4b
MKY	monkeyNFT0325/MKYFeedV1.sol	897f5de2a27ffb3b62dd17a041293547c752bbbd8b6334a0272f78d5faa16a89
MNF	monkeyNFT0325/Market.sol	790bc1a9ec55d32e7e57c5bb00d8a181f42e6c033db6ee181d1f764e0eaaa6d0
MNT	monkeyNFT0325/MonkeyNFT.sol	29e9ac3bdf53817f6c31b342f2d849876d1579b58858aa7c69c871dc1ab8c55d
MTN	monkeyNFT0325/MonkeyToken.sol	dcee675e42881f110417dd6d6635d7e87973383d67b4fb1fe96b2def58e69e1f
SNF	monkeyNFT0325/Strategy.sol	494527c18b713a97c933fce2e77947f2f186c45053c63a74ceb9300bd2f54943

Centralization

There are three roles (`DEFAULT_ADMIN_ROLE`/`WITHDRAW_ROLE`/`PAUSER_ROLE`) in this project. And a timelock will be added to `WITHDRAW_ROLE`.

- `DEFAULT_ADMIN_ROLE` can update settings through functions `updatedailyMKYRewardLimit`, `updateStrategy`, `updateStrategy()`, `updatePrice()` and `updatedailyMKYReward()`. Besides that, this role can recalculate rewards for the previous day through function `recountDailyReward()`.
- `WITHDRAW_ROLE` can transfer any amount of `tokenaddress` assets to any addresses through functions `withdraw()` and `withdrawETH()`.
- `PAUSER_ROLE` can update settings through functions `updatePause()`,`updateFeedPause()` and `updateRewardPause()`.

Findings



🟥	**Critical**	**0** (0.00%)
🟧	**Major**	**1** (5.00%)
⬜	**Minor**	**8** (40.00%)
🟦	**Informational**	**11** (55.00%)
🟩	**Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
MNF-1	Unprotected Access to Cancel Function	Control Flow	● Major	✓ Resolved
MNF-2	Missing Some Important Checks	Volatile Code	● Minor	✓ Resolved
MKY-1	Missing Some Important Checks	Volatile Code	● Minor	✓ Resolved
FVN-1	Missing Some Important Checks	Volatile Code	● Minor	✓ Resolved
BVN-1	Missing Some Important Checks	Volatile Code	● Minor	✓ Resolved
MKY-2	Missing Emit Event	Volatile Code	● Minor	✓ Resolved
FVN-2	Missing Emit Event	Volatile Code	● Minor	✓ Resolved
BVN-2	Missing Emit Event	Volatile Code	● Minor	✓ Resolved
MNF-3	Missing Emit Event	Volatile Code	● Minor	✓ Resolved
SNF-1	Unused Function Parameter	Coding Style	● Informational	✓ Resolved
MNF-4	Boolean Equality	Coding Style	● Informational	✓ Resolved
MKY-3	Missing Emit to Call Events	Coding Style	● Informational	✓ Resolved
FVN-3	Missing Emit to Call Events	Coding Style	● Informational	✓ Resolved
FVN-4	Error Warning Message	Coding Style	● Informational	✓ Resolved
MKY-4	Error Warning Message	Coding Style	● Informational	✓ Resolved
BVN-3	Error Warning Message	Coding Style	● Informational	✓ Resolved

ID	Title	Category	Severity	Status
MNF-5	Error Warning Message	Coding Style	● Informational	✓ Resolved
SNF-2	Feed Strategy in Strategy.sol	Logical Issue	● Informational	✓ Resolved
BVN-4	Hard Code Address	Logical Issue	● Informational	✓ Resolved
BVN-5	Proper Usage of payable in BoxesV1.sol	Coding Style	● Informational	✓ Resolved

MNF-1 | Unprotected Access to Cancel Function

Category	Severity	Location	Status
Control Flow	● Major	monkeyNFT0325/Market.sol: 96~104	✓ Resolved

Description

Any external users can cancel any order without identity check.

Recommendation

Consider adding additional check to make sure only nftOwner can call function `cancel()` .

```
require(nftOwner == msg.sender, "Only seller can cancel order");
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

MNF-2 | Missing Some Important Checks

Category	Severity	Location	Status
Volatile Code	● Minor	monkeyNFT0325/Market.sol: 145, 150	✓ Resolved

Description

Function `withdraw()` and `withdrawETH()` on the aforementioned lines is missing parameter address zero check.

Recommendation

Consider adding check to the two fuctions, for example:

```
function withdrawETH(address to) public {
      require(to != address(0), "withdraw-address-required");
      ...
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

MKY-1 | Missing Some Important Checks

Category	Severity	Location	Status
Volatile Code	● Minor	monkeyNFT0325/MKYFeedV1.sol: 201, 206	✓ Resolved

Description

Function `withdraw()` and `withdrawETH()` on the aforementioned lines is missing parameter address zero check.

Recommendation

Consider adding check to the two fuctions, for example:

```
function withdrawETH(address to) public {
     require(to != address(0), "withdraw-address-required");
     ...
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

FVN-1 | Missing Some Important Checks

Category	Severity	Location	Status
Volatile Code	● Minor	monkeyNFT0325/FeedV1.sol: 281, 286	✓ Resolved

Description

Function `withdraw()` and `withdrawETH()` on the aforementioned lines is missing parameter address zero check.

Recommendation

Consider adding check to the two fuctions, for example:

```
function withdrawETH(address to) public {
      require(to != address(0), "withdraw-address-required");
      ...
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

BVN-1 | Missing Some Important Checks

Category	Severity	Location	Status
Volatile Code	● Minor	monkeyNFT0325/BoxesV1.sol: 113, 118	✓ Resolved

Description

Function `withdraw()` and `withdrawETH()` on the aforementioned lines is missing parameter address zero check.

Recommendation

Consider adding check to the two fuctions, for example:

```
function withdrawETH(address to) public {
      require(to != address(0), "withdraw-address-required");
      ...
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

MKY-2 | Missing Emit Event

Category	Severity	Location	Status
Volatile Code	● Minor	monkeyNFT0325/MKYFeedV1.sol: 68	✓ Resolved

Description

It's sensitive to change the fee rate but no events are emited.

Function `updatePrice()` in contract `BoxesV1` and `FeedV1`.

Function `updatedailyMKYRewardLimit()` in contract `MKYFeedV1` and `FeedV1`.

Function `setfees()` in contract `Market`.

Recommendation

Consider emitting events when performing sensitive actions.

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

FVN-2 | Missing Emit Event

Category	Severity	Location	Status
Volatile Code	● Minor	monkeyNFT0325/FeedV1.sol: 90, 85	✓ Resolved

Description

It’s sensitive to change the fee rate but no events are emited.

Function `updatePrice()` in contract `BoxesV1` and `FeedV1`.

Function `updatedailyMKYRewardLimit()` in contract `MKYFeedV1` and `FeedV1`.

Function `setfees()` in contract `Market`.

Recommendation

Consider emitting events when performing sensitive actions.

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

BVN-2 | Missing Emit Event

Category	Severity	Location	Status
Volatile Code	● Minor	monkeyNFT0325/BoxesV1.sol: 67, 67	✓ Resolved

Description

It’s sensitive to change the fee rate but no events are emited.

Function `updatePrice()` in contract `BoxesV1` and `FeedV1`.

Function `updatedailyMKYRewardLimit()` in contract `MKYFeedV1` and `FeedV1`.

Function `setfees()` in contract `Market`.

Recommendation

Consider emitting events when performing sensitive actions.

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

MNF-3 | Missing Emit Event

Category	Severity	Location	Status
Volatile Code	● Minor	monkeyNFT0325/Market.sol: 140	✓ Resolved

Description

It’s sensitive to change the fee rate but no events are emited.

Function `updatePrice()` in contract `BoxesV1` and `FeedV1`.

Function `updatedailyMKYRewardLimit()` in contract `MKYFeedV1` and `FeedV1`.

Function `setfees()` in contract `Market`.

Recommendation

Consider emitting events when performing sensitive actions.

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

SNF-1 | Unused Function Parameter

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/Strategy.sol: 68, 51	✓ Resolved

Description

There parameters (`generation/mininggift/growthValue`) are not used in fuction `MKTFeedGetScore()`;

Parameter `growthValue` is not used in fuction `getScore()`.

Recommendation

Consider removing or commenting out the variable .

Alleviation

The development team responded that these variables will use in the future and have added comments.

```
//maybe growthValue will be used in the future.
function getScore...
...
//maybe generation , mininggift, growthValue will be used in the future.
function MKTFeedGetScore ...
```

MNF-4 | Boolean Equality

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/Market.sol: 61, 82	✓ Resolved

Description

Boolean constants can be used directly and do not need to be compare to true or false.

Recommendation

Consider removing the equality to the boolean constant.

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

MKY-3 | Missing Emit to Call Events

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/MKYFeedV1.sol: 135	✓ Resolved

Description

Missing `emit` when calling `MKYFEED` event in contract `MKYFeedV1`;

Missing `emit`when calling `FEED` event in contract `FeedV1`;

Recommendation

Consider adding `emit` to call events. For example:

```
        ....
        emit MKYFEED(....);
        ....
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

FVN-3 | Missing Emit to Call Events

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/FeedV1.sol: 174	✓ Resolved

Description

Missing `emit` when calling `MKYFEED` event in contract `MKYFeedV1`;

Missing `emit`when calling `FEED` event in contract `FeedV1`;

Recommendation

Consider adding `emit` to call events. For example:

```
    ....
   emit MKYFEED(....);
   ....
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

FVN-4 | Error Warning Message

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/FeedV1.sol: 86, 91, 96, 208, 282, 287, 86, 91, 106, 111	✓ Resolved

Description

Warning message is not correct.

Recommendation

Consider changing to the correct message， for example：

Function `updatedailyMKYRewardLimit()` and `withdraw()` in contract `MKYFeedV1`

```
 function updatedailyMKYRewardLimit(uint256 _toUpdatedailyMKYRewardLimit) external {
        require(hasRole(PAUSER_ROLE, _msgSender()), "MKYFeedV1: must have pauser role to
updatedailyMKYRewardLimit");
        dailyMKYRewardLimit=_toUpdatedailyMKYRewardLimit;
 }
...
function withdraw(address tokenaddress,address to) public {
        require(hasRole(WITHDRAW_ROLE, _msgSender()), "MKYFeedV1: must have withdraw
role to withdraw");
        IERC20(tokenaddress).transfer(to,IERC20(tokenaddress).balanceOf(address(this)));
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

MKY-4 | Error Warning Message

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/MKYFeedV1.sol: 202, 207, 69, 85, 90	✓ Resolved

Description

Warning message is not correct.

Recommendation

Consider changing to the correct message,　for example：

Function `updatedailyMKYRewardLimit()` and `withdraw()` in contract `MKYFeedV1`

```
 function updatedailyMKYRewardLimit(uint256 _toUpdatedailyMKYRewardLimit) external {
        require(hasRole(PAUSER_ROLE, _msgSender()), "MKYFeedV1: must have pauser role to
updatedailyMKYRewardLimit");
        dailyMKYRewardLimit=_toUpdatedailyMKYRewardLimit;
 }
...
function withdraw(address tokenaddress,address to) public {
        require(hasRole(WITHDRAW_ROLE, _msgSender()), "MKYFeedV1: must have withdraw
role to withdraw");
        IERC20(tokenaddress).transfer(to,IERC20(tokenaddress).balanceOf(address(this)));
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

BVN-3 | Error Warning Message

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/BoxesV1.sol: 68	✓ Resolved

Description

Warning message is not correct.

Recommendation

Consider changing to the correct message， for example：

Function `updatedailyMKYRewardLimit()` and `withdraw()` in contract `MKYFeedV1`

```
 function updatedailyMKYRewardLimit(uint256 _toUpdatedailyMKYRewardLimit) external {
        require(hasRole(PAUSER_ROLE, _msgSender()), "MKYFeedV1: must have pauser role to
updatedailyMKYRewardLimit");
        dailyMKYRewardLimit=_toUpdatedailyMKYRewardLimit;
 }
...
function withdraw(address tokenaddress,address to) public {
        require(hasRole(WITHDRAW_ROLE, _msgSender()), "MKYFeedV1: must have withdraw
role to withdraw");
        IERC20(tokenaddress).transfer(to,IERC20(tokenaddress).balanceOf(address(this)));
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

MNF-5 | Error Warning Message

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/Market.sol: 141, 146, 151	✓ Resolved

Description

Warning message is not correct.

Recommendation

Consider changing to the correct message， for example：

Function `updatedailyMKYRewardLimit()` and `withdraw()` in contract `MKYFeedV1`

```
 function updatedailyMKYRewardLimit(uint256 _toUpdatedailyMKYRewardLimit) external {
        require(hasRole(PAUSER_ROLE, _msgSender()), "MKYFeedV1: must have pauser role to updatedailyMKYRewardLimit");
        dailyMKYRewardLimit=_toUpdatedailyMKYRewardLimit;
 }
...
function withdraw(address tokenaddress,address to) public {
        require(hasRole(WITHDRAW_ROLE, _msgSender()), "MKYFeedV1: must have withdraw role to withdraw");
        IERC20(tokenaddress).transfer(to,IERC20(tokenaddress).balanceOf(address(this)));
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

SNF-2 | Feed Strategy in Strategy.sol

Category	Severity	Location	Status
Logical Issue	● Informational	monkeyNFT0325/Strategy.sol: 43~47	✓ Resolved

Description

Currently only one of them (`weightGrowth/mininggiftGrowth`) can be increased at the same time.

Alleviation

The development team responded that they don't want to improve both at the same time.

BVN-4 | Hard Code Address

Category	Severity	Location	Status
Logical Issue	● Informational	monkeyNFT0325/BoxesV1.sol: 39	✓ Resolved

Description

There are many hard code addresses in `FeedV1`, `BoxesV1`, `Market` and `MKYFeedV1`.

Alleviation

The development team responded that the `zero` address in contract `BoxesV1` just shows how many boxes have used, other addresses are their testing addresses.

BVN-5 | Proper Usage of payable in BoxesV1.sol

Category	Severity	Location	Status
Coding Style	● Informational	monkeyNFT0325/BoxesV1.sol: 77	✓ Resolved

Description

`payable` is used to receive the native token, but user can't withdraw them.

Recommendation

Consider removing `payable` from function `buy()`.

Alleviation

The development team heeded our advice and resolved this issue in commit `8702c0ff1491bb0a7813da9231c72db734d5538f`.

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.