Public

SMART CONTRACT AUDIT REPORT

for

MONKEY KING DEFIS

Hangzhou,China

September 28, 2021

Document Properties

Client	Monkey King Defis
Title	Smart contract audit report
Target	MIKE
Version	2.0
Author	Xu Jiang
Reviewed by	Jeff Liu
Approved by	Marsuye
Classification	Public

Version Info

Version	Date	Author(s)	Description
2.0	9/28, 2021	Xu Jiang	Final Release
1.8	8/02, 2021	Xu Jiang	Candidate
1.5+	6/20, 2021	Xu Jiang	Additional
1.2	3/07,2021	Xu Jiang	Biuding
1.1	12/22,2020	Xu Jiang	Accention
1.0	9/29,2020	Xu Jiang	Initial Draft

Block Audit Report Team received the MIKE team's application for smart contract security
audit of the MONKEY KING Token on September 28, 2021. The following are the details and results
of this smart contract security audit:

Token Name: MONKEY KING

The Contract address: 0x0100E08F5bc30e83B05c1C543F63961E0a376BB2

Link Address:
https://bscscan.com/address/0x0100e08f5bc30e83b05c1c543f63961e0a376bb2#code

The audit items and results:
(Other unknown security vulnerabilities are not included in the audit responsibility scope)

Audit Result: Passed
Audit Number: BAR0053928092021
Audit Date: September 28, 2021
Audit Team: Block Audit Report Team

Contents

Introduction

This Audit Report mainly focuses on the overall security of FREEDOM Token Smart Contract. With this
report, we have tried to ensure the reliability and correctness of their smart contract by complete and
rigorous assessment of their system's architecture and the smart contract codebase.

Auditing Approach and Methodologies applied

The Block Audit Report team has performed rigorous testing of the project starting with analyzing the

In the Unit testing Phase, we coded/conducted custom unit tests written for each function in the
vulnerabilities and security flaws.
The code was tested in collaboration of our multiple team members and this included -

● Testing the functionality of the Smart Contract to determine proper logic has been followed
throughout the whole process.
● Analyzing the complexity of the code in depth and detailed, manual review of the code, lineby-line.
● Checking whether all the libraries used in the code are on the latest version.
● Analyzing the security of the on-chain data.

Audit Details

Project Name: Monkey King
Website/ bscscan Code (Mainnet):
0x0100E08F5bc30e83B05c1C543F63961E0a376BB2
Languages: Solidity (Smart contract)
Platforms and Tools: Remix IDE, Truffle, Truffle Team, Ganache, Solhint, VScode, Mythril,

Contract.

Audit Goals

The focus of the audit was to verify that the Smart Contract System is secure, resilient and working
according to the specifications. The audit activities can be grouped in the following three categories:

Security

Identifying security related issues within each contract and the system of contract.

Sound Architecture

A full review of the contract source code. The primary areas of focus include:

- Accuracy
- Readability
- Sections of code with high complexity
- Quantity and quality of test coverage

High level severity issues

Issues on this level are critical to the smart contract's performance/functionality and should be fixed
before moving to a live environment.
Medium level severity issues
1. Issues on this level could potentially bring problems and should eventually be fixed.

Low level severity issues

Issues on this level are minor details and warningJs that can remain unfixed but would be better.

Issues Checking Status

NO	Issue description.	Checking status
1	Compiler warnings.	Passde
2	Possible delays in data delivery.	Passde
3	Oracle calls.	Passde
4	Front running.	Passde
5	Timestamp dependence.	Passde
6	Integer Overflow and Underflow.	Passde
7	DoS with Revert.	Passde
8	DoS with block gas limit.	Passde
9	Methods execution permissions.	Passde
10	Economy model.	Passde
11	The impact of the exchange rate	Passde
12	Private user data leaks.	Passde
13	Malicious Event log.	Passde
14	Scoping and Declarations.	Passde
15	Uninitialized storage pointers.	Passde
16	Arithmetic accuracy.	Passde

Used Code from other Framework/Smart Contracts (direct import)

[+] interface IERC20
- totalSupply()
- balanceOf(address account)
- transfer(address recipient, ...
- allowance(address owner, add ...
- approve(address spender, uin ...
- transferFrom(address sender, ...

[+] library SafeMath
- add(uint256 a, uint256 b)
- sub(uint256 a, uint256 b)
- sub(uint256 a, uint256 b, st ...
- mul(uint256 a, uint256 b)
- div(uint256 a, uint256 b)
- div(uint256 a, uint256 b, st ...
- mod(uint256 a, uint256 b)
- mod(uint256 a, uint256 b, st ...
- msgSender()
- msgData()

[+] library Address
- isContract(address account)
- sendValue(address payable re ...
- Call(address target, ...
- Call(address target, ...
- CallWithValue(addres ...
- CallWithValue(addres ...
- CallWithValue(addre ...

[+] contract Ownable is Context
- owner()
- renounceOwnership()

- geUnlockTime()
- lock(uint256 time)
- unlock()

[+] interface IUniswapV2Factory

- feeTo()
- feeToSetter()
- getPair(address tokenA, addr ...
- allPairs(uint)
- allPairsLength()
- createPair(address tokenA, a ...
- setFeeTo(address)
- setFeeToSetter(address)

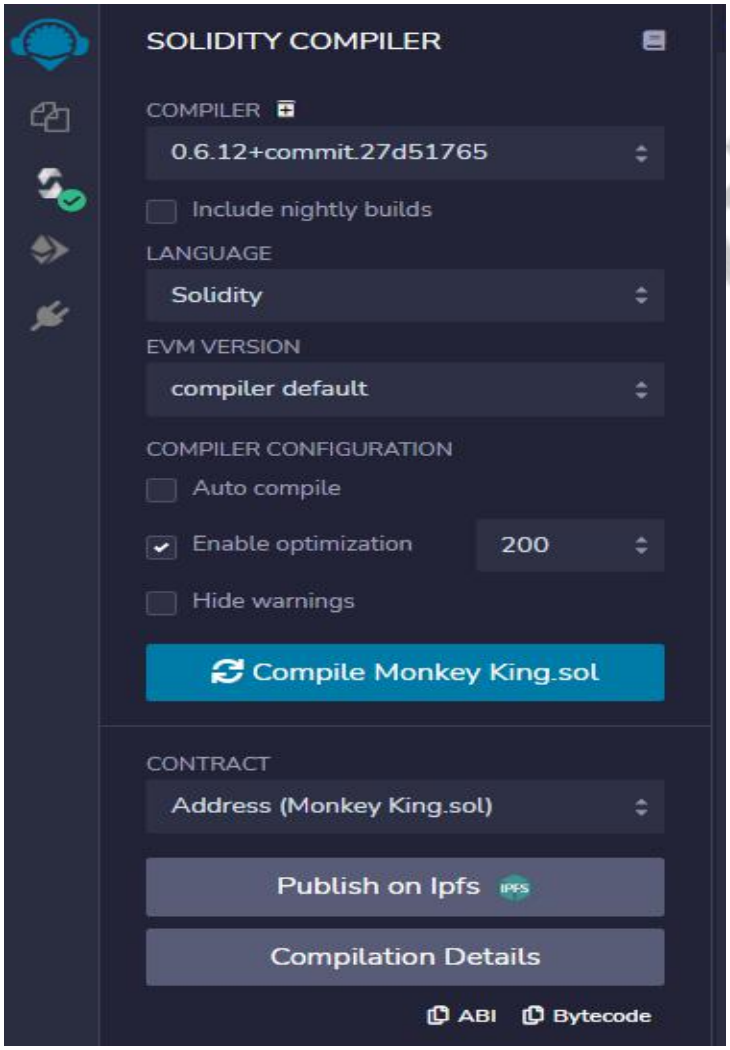
[+] interface IUniswapV2Pair

- name()
- symbol()
- decimals()
- totalSupply()
- balanceOf(address owner)
- allowance(address owner, add ...
- approve(address spender, uin ...
- transfer(address to, uint va ...
- transferFrom(address from, a ...
- DOMAINSEPARATOR()
- PERMITTYPEHASH()
- nonces(address owner)
- permit(address owner, addres ...
- MINIMUMLIQUIDITY()
- factory()
- token0()
- token1()
- getReserves()
- price0CumulativeLast()

Automated Audit

Remix Compiler Warnings

It throws warnings by Solidity's compiler. If it encounters any errors the contract cannot be
compiled and deployed.



Freedom.sol: Warning: SPDX license identifier not provided in source file. Before publishing, consider
adding a comment containing "SPDX-License-Identifier: <SPDX-License>" to each source file.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice
as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and
algorithms based on smart contracts, the details of which are set out in this report. In order to get a full
view of our analysis, it is crucial for you to read the full report. While we have done our best in
conducting our analysis and producing this report, it is important to note that you should not rely on
this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced
it, and it is important for you to conduct your own independent investigations before making any
decisions. We go into more detail on this in the below disclaimer below – please make sure to read it
in full.
DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do
not agree to the terms, then please immediately cease reading this report, and delete and destroy any
and all copies of this report downloaded and/or printed by you. This report is provided for information
purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall
have any right to rely on the report or its contents, and BlockAudit and its affiliates (including holding
companies, shareholders, subsidiaries, employees, directors, officers and other representatives)
(BlockAudit) owe no duty of care towards you or any other person, nor does BlockAudit make any
warranty or representation to any person on the accuracy or completeness of the report. The report is
provided "as is", without any conditions, warranties or other terms of any kind

Summary

except as set out in this
disclaimer, and BlockAudit hereby excludes all representations, warranties, conditions and other terms
(including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose
and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the
report. Except and only to the extent that it is prohibited by law, BlockAudit hereby excludes all liability
and responsibility, and neither you nor any other person shall have any claim against BlockAudit, for
any amount or kind of loss or damage that may result to you or any other person (including without
limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or
any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and
whether in delict, tort (including without limitation negligence), contract, breach of statutory duty,
misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature
whatsoever in any jurisdiction) in any way arising from or connected with this report and the use,
inability to use or the results of use of this report, and any reliance on this report.
The analysis of the security is purely based on the smart contracts alone. No applications or operations
were reviewed for security. No product code has been reviewed.
14
Smart contracts do not contain any high severity issues!
Note:
Please check the disclaimer above and note, the audit makes no statements or warranties on business
model, investment attractiveness or code sustainability. The report is provided.

Official website

www.monkeyking.buzz

E-Mail

team@monkeyking.com

Twitter

https://twitter.com/MIKE_webs