



Smart Contract Audit

FOR

Moonerium

DATED : 31 October 23'



AUDIT SUMMARY

Project name – Moonerium

Date: 31 October 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: Passed

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	3
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1- Manual Review:

A line by line code review has been performed by audit ace team.

2- BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3- Slither :

The code has undergone static analysis using Slither.

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/token/0x11E9318A6FBD162a752db8eAbD7E93635b7B220E>



Token Information

Token Address :

0x0CEd57C3E4Bd6F40f333134c93fB5143b30d31ad

Name: Moonerium

Symbol: MOONERIUM

Decimals: 18

Network: Ethereum mainnet

Token Type: ERC20

Owner: 0x4494cB7c29E663CF469e9c478f2DD989A2b5F0e9

Deployer:

0x4494cB7c29E663CF469e9c478f2DD989A2b5F0e9

Token Supply: 1,000,000,000

Checksum:

af747e29e250fa2181f56bced993ee804a62665c

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:
<https://testnet.bscscan.com/token/0x11E9318A6FBD162a752db8eAbD7E93635b7B220E>



TOKEN OVERVIEW

buy fee: 0-0.1%

Sell fee: 0-0.1%

transfer fee: 0%

Fee Privilege: Static fees

Ownership: Owned

Minting: None

Max Tx: No

Blacklist: No

Other Privileges:

- Initial distribution of the tokens



AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
 - Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
 - Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
 - Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
 - Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
 - Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.
-



VULNERABILITY CHECKLIST

- | | |
|------------------------------------|-------------------------------|
| ✓ Return values of low-level calls | ✓ Gasless Send |
| ✓ Private modifier | ✓ Using block.timestamp |
| ✓ Multiple Sends | ✓ Re-entrancy |
| ✓ Using Suicide | ✓ Tautology or contradiction |
| ✓ Gas Limitand Loops | ✓ Timestamp Dependence |
| ✓ Address hardcoded | ✓ Revert/require functions |
| ✓ Exception Disorder | ✓ Use of tx.origin |
| ✓ Using inline assembly | ✓ Integer overflow/underflow |
| ✓ Divide before multiply | ✓ Dangerous strict equalities |
| ✓ Missing Zero Address Validation | ✓ Using SHA3 |
| ✓ Compiler version not fixed | ✓ Using throw |
-



CLASSIFICATION OF RISK

Severity

Description

◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization / Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

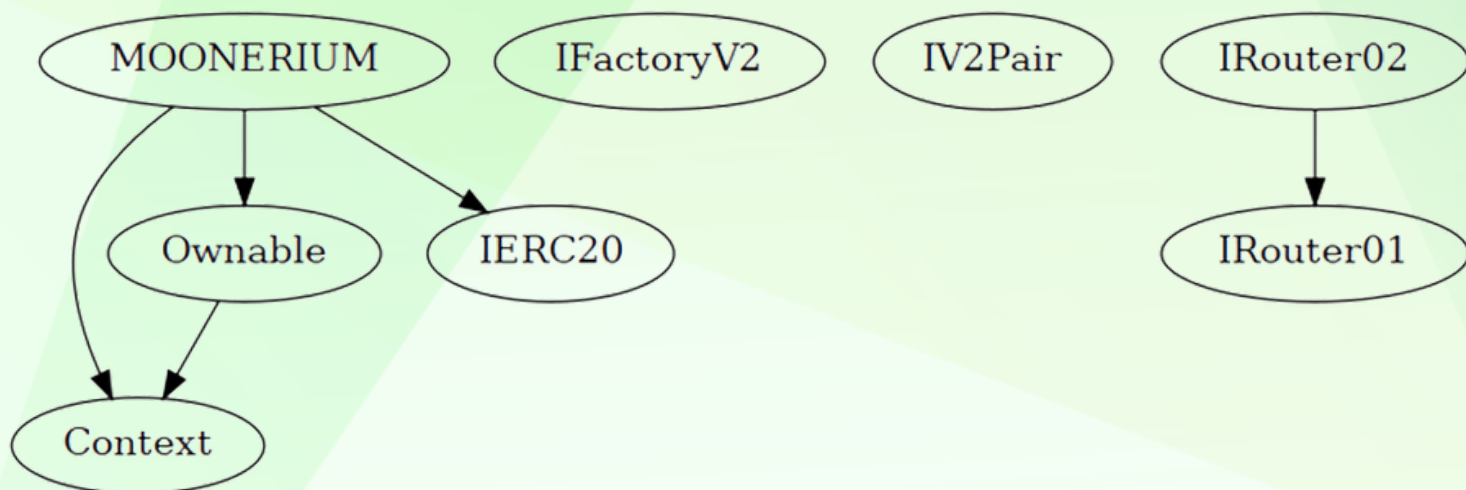
Findings

Severity

Found

◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	3

INHERITANCE TREE





GENERAL FINDINGS SUMMARY

- **Buy Fees** : 0.1% (Static)
 - **Sell Fees** : 0.1% (Static)
 - **Transfer Fees** : 0% (Static)
 - **Blacklisting**: no
 - **Disabling trades** : no
 - **Minting new tokens**: no
 - **Maximum wallet/swap/transfer limitation**: no
-



STATIC ANALYSIS

```
INFO:Detectors:
Context._msgData() (contracts/Token.sol#22-25) is never used and should be removed
MOONERIUM.isLimitedAddress(address,address) (contracts/Token.sol#404-418) is never used and should be removed
MOONERIUM.is_transfer(address,address) (contracts/Token.sol#430-436) is never used and should be removed
MOONERIUM.swapAndLiquify(uint256) (contracts/Token.sol#510-554) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.17 (contracts/Token.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in MOONERIUM.internalSwap(uint256) (contracts/Token.sol#556-591):
- (success,None) = marketingAddress.call(gas: 35000,value: marketingAmount)() (contracts/Token.sol#584-586)
- (success,None) = developmentAddress.call(gas: 35000,value: developmentAmount)() (contracts/Token.sol#587-590)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IRouter01.WETH() (contracts/Token.sol#102) is not in mixedCase
Event MOONERIUM._setPresaleAddress(address,bool) (contracts/Token.sol#314) is not in CapWords
Event MOONERIUM._toggleCanSwapFees(bool) (contracts/Token.sol#315) is not in CapWords
Event MOONERIUM._changePair(address) (contracts/Token.sol#316) is not in CapWords
Event MOONERIUM._changeThreshold(uint256) (contracts/Token.sol#317) is not in CapWords
Event MOONERIUM._changeWalleets(address) (contracts/Token.sol#318) is not in CapWords
Event MOONERIUM._changeFees(uint256,uint256) (contracts/Token.sol#319) is not in CapWords
Function MOONERIUM.is_buy(address,address) (contracts/Token.sol#420-423) is not in mixedCase
Function MOONERIUM.is_sell(address,address) (contracts/Token.sol#425-428) is not in mixedCase
Function MOONERIUM.is_transfer(address,address) (contracts/Token.sol#430-436) is not in mixedCase
Constant MOONERIUM.buyfee (contracts/Token.sol#266) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOONERIUM.sellfee (contracts/Token.sol#267) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOONERIUM.transferfee (contracts/Token.sol#268) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOONERIUM.marketingFee (contracts/Token.sol#270) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOONERIUM.developmentFee (contracts/Token.sol#271) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOONERIUM._name (contracts/Token.sol#296) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOONERIUM._symbol (contracts/Token.sol#297) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOONERIUM._decimals (contracts/Token.sol#302) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (contracts/Token.sol#23)" inContext (contracts/Token.sol#15-26)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
Variable IRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#119) is too similar to IRouter01.addLiquidity(
address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#120)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
MOONERIUM.buyAllocation (contracts/Token.sol#292) should be constant
MOONERIUM.developmentAddress (contracts/Token.sol#288-289) should be constant
MOONERIUM.marketingAddress (contracts/Token.sol#286-287) should be constant
MOONERIUM.sellAllocation (contracts/Token.sol#293) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
MOONERIUM.lpPair (contracts/Token.sol#304) should be immutable
MOONERIUM.swapRouter (contracts/Token.sol#295) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Token.sol analyzed (8 contracts with 88 detectors), 38 result(s) found
```

**Result => A static analysis of contract's source code has
been performed using slither,
No major issues were found in the output**



CONTRACT ASSESMENT

```
| Contract | Type | Bases | | |
|:-----:|:-----:|:-----:|:-----:|:-----:|
|   ↳   | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
|||||
| **Context** | Implementation ||||
|   ↳ | <Constructor> | Public ! | ● | NO ! |
|   ↳ | _msgSender | Internal 🔒 | | |
|   ↳ | _msgData | Internal 🔒 | | |
|||||
| **Ownable** | Implementation | Context |||
|   ↳ | <Constructor> | Public ! | ● | NO ! |
|   ↳ | owner | Public ! | NO ! |
|   ↳ | renounceOwnership | Public ! | ● | onlyOwner |
|   ↳ | transferOwnership | Public ! | ● | onlyOwner |
|   ↳ | _setOwner | Private 🔒 | ● | |
|||||
| **IFactoryV2** | Interface ||||
|   ↳ | getPair | External ! | NO ! |
|   ↳ | createPair | External ! | ● | NO ! |
|||||
| **IV2Pair** | Interface ||||
|   ↳ | factory | External ! | NO ! |
|   ↳ | getReserves | External ! | NO ! |
|   ↳ | sync | External ! | ● | NO ! |
|||||
| **IRouter01** | Interface ||||
|   ↳ | factory | External ! | NO ! |
|   ↳ | WETH | External ! | NO ! |
|   ↳ | addLiquidityETH | External ! | 💰 | NO ! |
|   ↳ | addLiquidity | External ! | ● | NO ! |
|   ↳ | swapExactETHForTokens | External ! | 💰 | NO ! |
|   ↳ | getAmountsOut | External ! | | NO ! |
|   ↳ | getAmountsIn | External ! | | NO ! |
|||||
```



CONTRACT ASSESMENT

| ****IRouter02**** | Interface | IRouter01 |||

| | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | ● |NO ! |

| | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | 🟢 |NO ! |

| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | ● |NO ! |

| | swapExactTokensForTokens | External ! | ● |NO ! |

|||||

| ****IERC20**** | Interface |||

| | totalSupply | External ! ||NO ! |

| | decimals | External ! ||NO ! |

| | symbol | External ! ||NO ! |

| | name | External ! ||NO ! |

| | getOwner | External ! ||NO ! |

| | balanceOf | External ! ||NO ! |

| | transfer | External ! | ● |NO ! |

| | allowance | External ! ||NO ! |

| | approve | External ! | ● |NO ! |

| | transferFrom | External ! | ● |NO ! |

|||||

| ****MOONERIUM**** | Implementation | Context, Ownable, IERC20 |||

| | totalSupply | External ! ||NO ! |

| | decimals | External ! ||NO ! |

| | symbol | External ! ||NO ! |

| | name | External ! ||NO ! |

| | getOwner | External ! ||NO ! |

| | allowance | External ! ||NO ! |

| | balanceOf | Public ! ||NO ! |

| | viewTaxes | External ! ||NO ! |

| | <Constructor> | Public ! | ● |NO ! |

| | <Receive Ether> | External ! | 🟢 |NO ! |

| | transfer | Public ! | ● |NO ! |

| | approve | External ! | ● |NO ! |




| | _approve | Internal 🔒 | ● |



| | transferFrom | External ! | ● |NO ! |



| | isNoFeeWallet | External ! | |NO ! |







CONTRACT ASSESMENT



|  | setNoFeeWallet | Public  |  | onlyOwner |




|  | isLimitedAddress | Internal  | | |




|  | is_buy | Internal  | | |




|  | is_sell | Internal  | | |




|  | is_transfer | Internal  | | |




|  | canSwap | Internal  | | |




|  | changeLpPair | External  |  | onlyOwner |




|  | toggleCanSwapFees | External  |  | onlyOwner |

|  | _transfer | Internal  |  | |

|  | takeTaxes | Internal  |  | |

|  | swapAndLiquify | Internal  |  | inSwapFlag |


|  | internalSwap | Internal  |  | inSwapFlag |


|  | setPresaleAddress | External  |  | onlyOwner |

Legend

| Symbol | Meaning |

|:-----:|-----|

|  | Function can modify state |

|  | Function is payable |



FUNCTIONAL TESTING

◆ **Adding liquidity on PCS V2 (passed):**

<https://testnet.bscscan.com/tx/0x9f0db6b71315394173fc9d3b6bfd86e5633a3852622d745a59eec31ef1c28309>

◆ **BNB => Token swap from a whitelisted wallet (0% tax) (passed):**

<https://testnet.bscscan.com/tx/0x9f22d34c6df1f137232503b87534fcdac40369e534ee8d83ce8594f51acd7f88>

◆ **Token => BNB swap from a whitelisted wallet (0% tax) (passed):**

<https://testnet.bscscan.com/tx/0x3f1c84128b5ed6e4673760458fccb7e6b94b40b4f4e29ca615c27d678f0bc770>

◆ **Transferring from a whitelisted wallet (0% tax) (passed):**

<https://testnet.bscscan.com/tx/0xba5af24a02bac54c573356dd47ef52bc270222f3632cd335516b2ff18e91286>

◆ **BNB=>Token swap from a regular wallet (tax 0-0.1%) (passed):**

<https://testnet.bscscan.com/tx/0x9812de4a2c709a7d4df1875c5fcde184fabfc6907a254440c03e8c577c0bbbe1>

◆ **Token=>BNB swap from a regular wallet (tax 0-0.1%) (passed):**

<https://testnet.bscscan.com/tx/0x09b36f0a33cc65b32579dad4cf4ec1fc5dac4dcbead1a2017175859507ef70c5>

◆ **Transferring between 2 regular wallets (0% tax) (passed):**

<https://testnet.bscscan.com/tx/0xd886db3b2900ac5ec813bdd1d4a5cd7e6c4bc70d1d3cb21ba930a29d545a982a>

◆ **Internal swap of accumulated buy/sell fees (BNB sent to marketing and development wallets) (passed):**

<https://testnet.bscscan.com/tx/0x09b36f0a33cc65b32579dad4cf4ec1fc5dac4dcbead1a2017175859507ef70c5>

MANUAL TESTING

[NC-0] – Lack of event emission:

Issue: `setNoFeeWallet` is changing state of the contract but not emitting any events.

Mitigation: emit an event from `setNoFeeWallet`

```
function setNoFeeWallet(address account, bool enabled) public  
onlyOwner {  
    _noFee[account] = enabled;  
    emit NoFeeWalelt(account, enabled);  
}
```

=====

[NC-1] – Unused code:

Issue: `swapAndLiquify` and `isLimitedAddress` and their related variables and events are not used anywhere in the contract.

Mitigation: Delete `swapAndLiquify` and `isLimitedAddress` functions in order to reduce contract size and improve contract readability

=====

[NC-0] – Stuck ERC20 tokens and ETH:

Issue: contract implements a “`receive()`” function which means it accepts receiving BNB, but there are no functions to recover those BNB and ERC20 tokens **manually**.

Mitigation: Implement a function for withdrawing BNB and ERC20 tokens from the contract



DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.



ABOUT AUDITACE

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



<https://auditace.tech/>



https://t.me/Audit_Ace



https://twitter.com/auditace_



<https://github.com/Audit-Ace>
