auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

BuyBack70Floki

DATED : 16 March, 2024

AUDIT SUMMARY

Project name – BuyBack70Floki

Date: 16 March, 2024

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

Audit Status: **Passed with high risk**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	1	0	2	1
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/address/0xECdF05B54D0B37aa4B04e2371b74438725150181#code

Token Information

Token Name : BuyBack70Floki

Token Symbol: BUYBACK70

Decimals: 18

Token Supply: 70000

Network: BscScan

Token Type: BEP-20

Token Address:
0xF65A4183fe9E331122CF279d7A72D99E1452f629

Checksum:
B2032c616934aeb47e6039f76b20d251

Owner:
0x6b8F50E04079C408a67ec8a016133E7e0FDa6E81
(at time of writing the audit)

Deployer:
0x6b8F50E04079C408a67ec8a016133E7e0FDa6E81

TOKEN OVERVIEW

Fees:

Buy Fee: 4%

Sell Fee: 4%

Transfer Fee: 0%

Fees Privilege: Owner

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

AUDIT METHODOLOGY

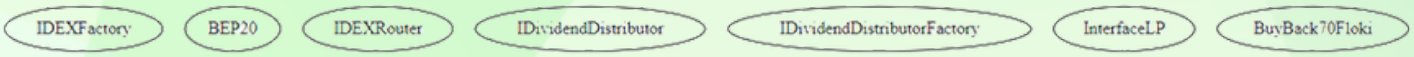
The auditing process will follow a routine as special considerations by Auditace:

- 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 isexercised 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

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

INHERITANCE TREE



STATIC ANALYSIS

A static analysis of the code was performed using Slither.

No issues were found.

```
        swapEnabled = True
Traceback (most recent call last):
  File "/home/ethsec/.local/bin/slither", line 8, in <module>
    sys.exit(main())
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/__main__.py", line 727, in main
    main_impl(all_detector_classes=detectors, all_printer_classes=printers)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/__main__.py", line 833, in main_impl
    ) = process_all(filename, args, detector_classes, printer_classes)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/__main__.py", line 107, in process_all
    ) = process_single(compilation, args, detector_classes, printer_classes)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/__main__.py", line 80, in process_single
    slither = Slither(target, ast_format=ast, **vars(args))
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/slither.py", line 144, in __init__
    self._init_parsing_and_analyses(kwargs.get("skip_analyze", False))
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/slither.py", line 164, in _init_parsing_and_analyses
    raise e
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/slither.py", line 160, in _init_parsing_and_analyses
    parser.analyze_contracts()
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/solc_parsing/slither_compilation_unit_solc.py", line 539, in analyze_contracts
    self._convert_to_slithir()
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/solc_parsing/slither_compilation_unit_solc.py", line 765, in _convert_to_slithir
    raise e
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/solc_parsing/slither_compilation_unit_solc.py", line 750, in _convert_to_slithir
    func.generate_slithir_and_analyze()
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/core/declarations/function.py", line 1767, in generate_slithir_and_analyze
    node.slithir_generation()
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/core/cfg/node.py", line 716, in slithir_generation
    self._irs = convert_expression(expression, self)  # type:ignore
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/slithir/convert.py", line 115, in convert_expression
    visitor = ExpressionToSlithIR(expression, node)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/visitors/slithir/expression_to_slithir.py", line 174, in __init__
    self._visit_expression(self.expression)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/visitors/expression/expression.py", line 43, in _visit_expression
    self._visit_assignement_operation(expression)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/visitors/expression/expression.py", line 99, in _visit_assignement_operation
    self._visit_expression(expression.expression_right)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/visitors/expression/expression.py", line 52, in _visit_expression
    self._visit_conditional_expression(expression)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/visitors/expression/expression.py", line 119, in _visit_conditional_expression
    self._visit_expression(expression.else_expression)
  File "/home/ethsec/.local/lib/python3.10/site-packages/slither/visitors/expression/expression.py", line 52, in _visit_expression
```

FUNCTIONAL TESTING

1- Approve (passed):

https://testnet.bscscan.com/tx/0x4a9ccb8fff83e675105ed108e7334b28501678fd91e7b6cab54c81154832dbe3

2- Set marketing Fee Receiver (passed):

https://testnet.bscscan.com/tx/0x21c8e32a8cfd34481a0c36b96f5d40cff560ad43b06c6931bb64560a551f79b9

3- Set Reward Token (passed):

https://testnet.bscscan.com/tx/0x269aa448554f5a4fdaa563f596c9fd641e6602ceb17645a5b5edbc1118078f6e

4- Set Swap Back Settings (passed):

https://testnet.bscscan.com/tx/0x8c3006d01af803d2cce2c27059f252b45cd1e0b5665f1fea5bd6dd4e56a630c0

5- Set pair (passed):

https://testnet.bscscan.com/tx/0x2829441153d4d8ccd016b9789d6af743741df8c1e9d158abdfe4a402d4fd7a38

POINTS TO NOTE

- The owner can transfer ownership.
- The owner can renounce ownership.
- The owner can set a pair.
- The owner excludes/includes an address from fees.
- The owner set dividend exempt.
- The owner can set the marketing fee receiver address.
- The owner can set the reward token address.
- The owner enables swap-back settings.
- The owner can burn lp.
- The owner can manually send.
- The owner can set distributor to buy gas.

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	1
Medium-Risk	0
Low-Risk	2
Gas Optimization / Suggestions	1

MANUAL TESTING

Centralization – Missing Require Check

Severity: High

Function: setmarketingFeeReceiver/setRewardToken

Status: Open

Overview:

The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner sets the address to the contract address, then the ETH will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.

```
function setmarketingFeeReceiver(address _marketingFeeReceiver)
        external
        onlyOwner
    {
        marketingFeeReceiver = _marketingFeeReceiver;
    }
 function setRewardToken(address rewardToken) external onlyOwner {
    distributor = IDividendDistributor(
        distributorFactory.createDistribuitor(
            address(router),
            rewardToken
        )
    );
}
```

Suggestion:

It is recommended that the address should not be able to set as a contract address.

MANUAL TESTING

Centralization – Missing Events

Severity: Low

Subject: Missing Events

Status: Open

Overview:

They serve as a mechanism for emitting and recording data onto the blockchain, making it transparent and easily accessible.

```
function setPair(address addr, bool io) public onlyOwner {
        pair[addr] = io;
    }
function setDividendExempt(address account, bool b) public onlyOwner {
        isDividendExempt[account] = b;
    }
function setmarketingFeeReceiver(address _marketingFeeReceiver)
        external
        onlyOwner
    {
        marketingFeeReceiver = _marketingFeeReceiver;
    }
function setRewardToken(address rewardToken) external onlyOwner {
    distributor = IDividendDistributor(
        distributorFactory.createDistribuitor(
            address(router),
            rewardToken
        )
    );
}
```

Suggestion:

Emit an event for critical changes.

MANUAL TESTING

Centralization – Missing Visibility

Severity: Low

Subject: mapping

Status: Open

Overview:

It's simply saying that no visibility was specified, so it's going with the default. This has been related to security issues in contracts.

```
mapping(address => uint256) _balances;
mapping(address => mapping(address => uint256)) _allowances;
address[] quotesReceiver;
bool inSwap;
```

Suggestion:

You can easily silence the warning by adding the mapping public:

MANUAL TESTING

Optimization

Severity: Optimization

Subject: Remove unused code

Status: Open

Overview:

Unused variables are allowed in Solidity, and they do not pose a direct security issue. It is the best practice to avoid them.

```
interface BEP20 {
    function balanceOf(address account) external view returns (uint256);

    function transfer(address recipient, uint256 amount)
        external
        returns (bool);

    function approve(address spender, uint256 amount) external returns (bool);

    function transferFrom(
        address sender,
        address recipient,
        uint256 amount
    ) external returns (bool);

    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(
        address indexed owner,
        address indexed spender,
        uint256 value
    );
}
```

Suggestion:

To reduce high gas fees. It is suggested to remove unused code 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 specializes 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