auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

MOVIBIT

DATED : 12 May 23'

AUDIT SUMMARY

Project name – MOVIBIT

Date: 12 May, 2023

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	1	0	1	2	0
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/0x10E19de3285a7913aEEdDc402d92aBe7D9548FE4

Token Information

Token Name : MOVIBIT

Token Symbol: MVBT

Decimals: 5

Token Supply: 280,000,000

Token Address:
0x5d6Fe0FB6d8f44C1b41151D1Afff3b3ae68C8326

Checksum:
7b25aa02430e71f6b104fc5e6bf9418f3a8edbf8

Owner:
0x7E68dC4993007e7Cc1221e28fd83d964444eE93A

Deployer:
0x7E68dC4993007e7Cc1221e28fd83d964444eE93A

TOKEN OVERVIEW

Fees:

Buy Fees: up to 15%

Sell Fees: up to 15%

Transfer Fees: up to 15%

Fees Privilege: Owner

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: changing fee - including and excluding form fee - changing distribution settings (min tokens to be eligible, cool down between claims etc)

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

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

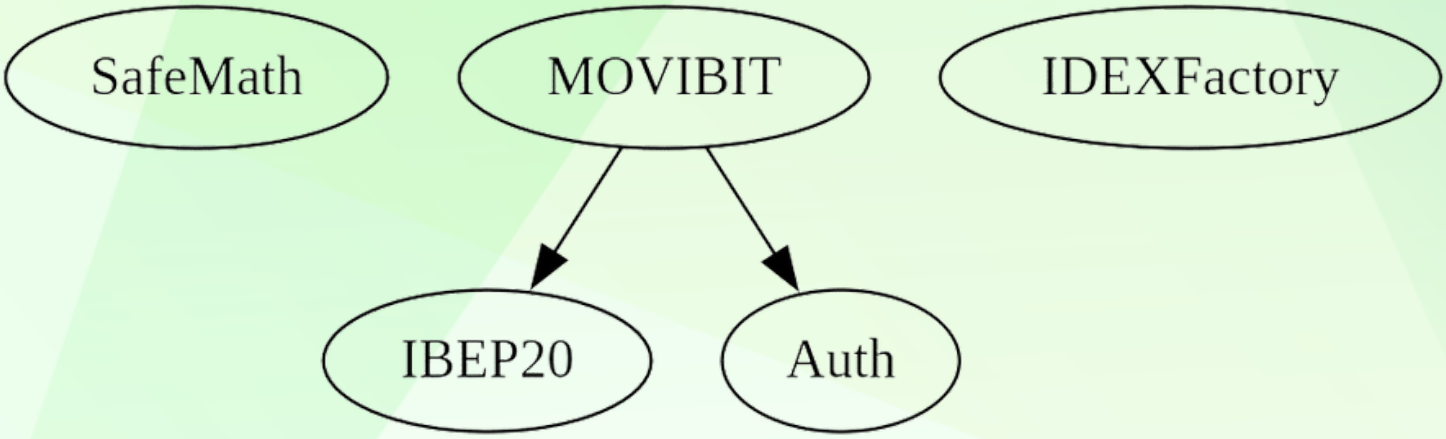
CLASSIFICATION OF RISK

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

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to set 15% fee for buy/sell/transfer each
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
IBEP20	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 ❗
Auth	Implementation			
└	<Constructor>	Public ❗	🛑	NO ❗
└	authorize	Public ❗	🛑	authorized
└	unauthorize	Public ❗	🛑	authorized
└	isOwner	Public ❗		NO ❗
└	isAuthorized	Public ❗		NO ❗
└	transferOwnership	Public ❗	🛑	onlyOwner
IDEXFactory	Interface			
└	createPair	External ❗	🛑	NO ❗
IDEXRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🛑	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗

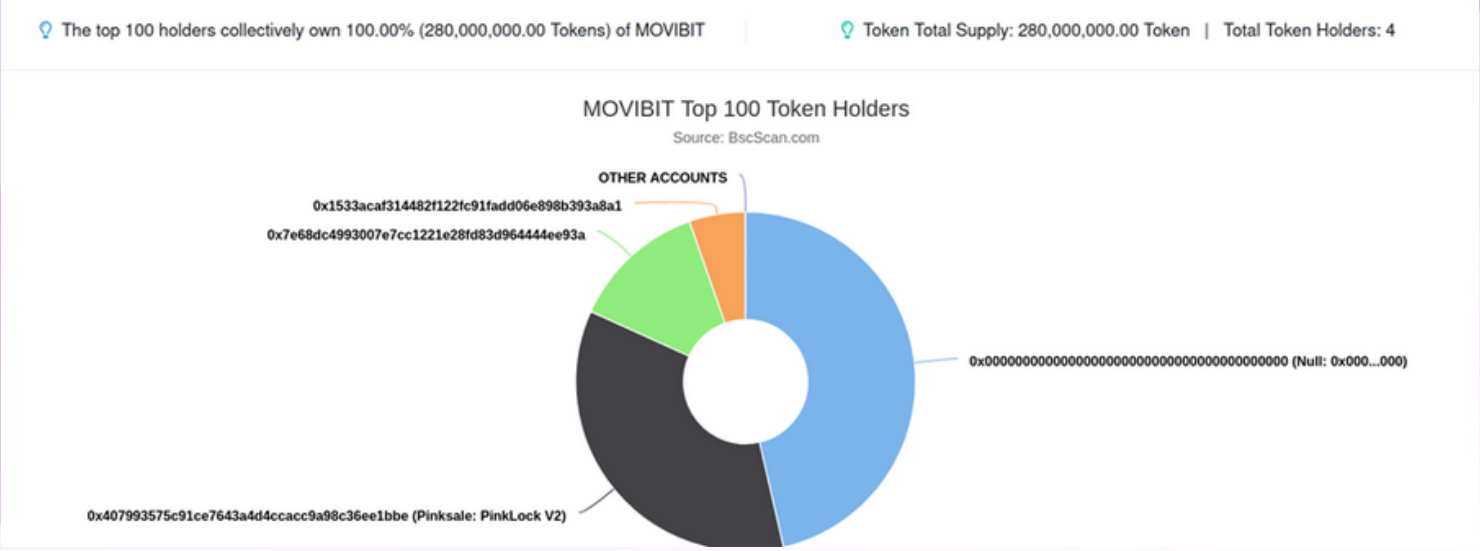
CONTRACT ASSESMENT

MOVIBIT	Implementation	IBEP20, Auth		
└	<Constructor>	Public ❗	🛑	Auth
└	<Receive Ether>	External ❗	💵	NO ❗
└	totalSupply	External ❗		NO ❗
└	decimals	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	name	External ❗		NO ❗
└	getOwner	External ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	allowance	External ❗		NO ❗
└	approve	Public ❗	🛑	NO ❗
└	approveMax	External ❗	🛑	NO ❗
└	transfer	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
└	burn	Public ❗	🛑	NO ❗
└	_transferFrom	Internal 🔒	🛑	
└	_basicTransfer	Internal 🔒	🛑	
└	shouldTakeFee	Internal 🔒		
└	takeFee	Internal 🔒	🛑	
└	shouldSwapBack	Internal 🔒		
└	saveStuckTokens	External ❗	🛑	authorized
└	saveStuckBNB	External ❗	🛑	authorized
└	isFreeFees	Public ❗		NO ❗
└	swapBack	Internal 🔒	🛑	swapping
└	setIsFeeExempt	External ❗	🛑	authorized
└	setFeesExchange	External ❗	🛑	authorized
└	setFeesTransfer	External ❗	🛑	authorized
└	setSwapBackSettings	External ❗	🛑	authorized
└	multiTransfer	External ❗	🛑	onlyOwner

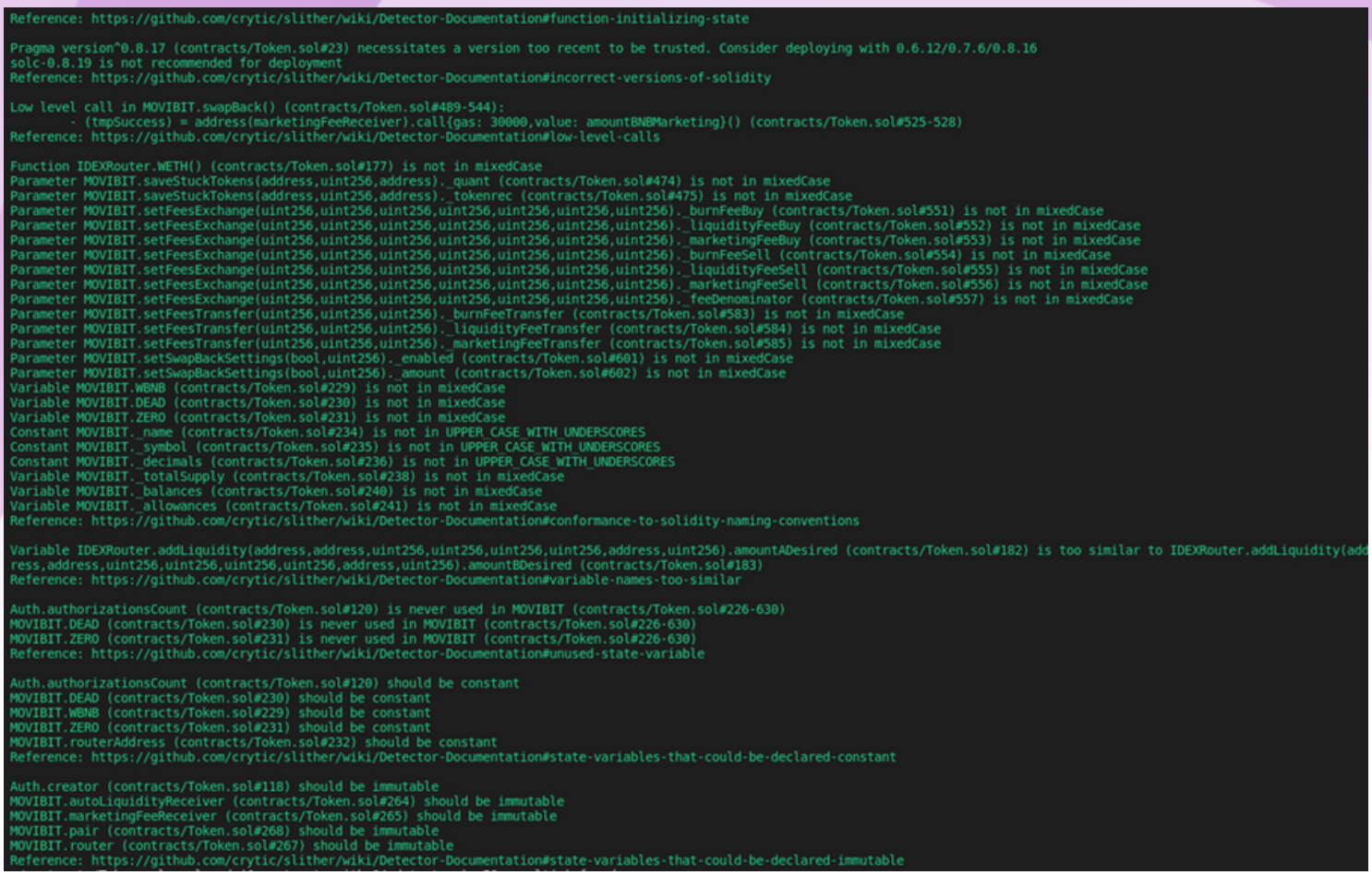
Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

TOKENOMICS AT TIME OF AUDIT



STATIC ANALYSIS



```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#23) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in MOVIBIT.swapBack() (contracts/Token.sol#489-544):
        - (tmpSuccess) = address(marketingFeeReceiver).call{gas: 30000,value: amountBNBMarketing}() (contracts/Token.sol#525-528)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IDEXRouter.WETH() (contracts/Token.sol#177) is not in mixedCase
Parameter MOVIBIT.saveStuckTokens(address,uint256,address)._quant (contracts/Token.sol#474) is not in mixedCase
Parameter MOVIBIT.saveStuckTokens(address,uint256,address)._tokenrec (contracts/Token.sol#475) is not in mixedCase
Parameter MOVIBIT.setFeesExchange(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._burnFeeBuy (contracts/Token.sol#551) is not in mixedCase
Parameter MOVIBIT.setFeesExchange(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._liquidityFeeBuy (contracts/Token.sol#552) is not in mixedCase
Parameter MOVIBIT.setFeesExchange(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._marketingFeeBuy (contracts/Token.sol#553) is not in mixedCase
Parameter MOVIBIT.setFeesExchange(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._burnFeeSell (contracts/Token.sol#554) is not in mixedCase
Parameter MOVIBIT.setFeesExchange(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._liquidityFeeSell (contracts/Token.sol#555) is not in mixedCase
Parameter MOVIBIT.setFeesExchange(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._marketingFeeSell (contracts/Token.sol#556) is not in mixedCase
Parameter MOVIBIT.setFeesExchange(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._feeDenominator (contracts/Token.sol#557) is not in mixedCase
Parameter MOVIBIT.setFeesTransfer(uint256,uint256,uint256)._burnFeeTransfer (contracts/Token.sol#583) is not in mixedCase
Parameter MOVIBIT.setFeesTransfer(uint256,uint256,uint256)._liquidityFeeTransfer (contracts/Token.sol#584) is not in mixedCase
Parameter MOVIBIT.setFeesTransfer(uint256,uint256,uint256)._marketingFeeTransfer (contracts/Token.sol#585) is not in mixedCase
Parameter MOVIBIT.setSwapBackSettings(bool,uint256)._enabled (contracts/Token.sol#601) is not in mixedCase
Parameter MOVIBIT.setSwapBackSettings(bool,uint256)._amount (contracts/Token.sol#602) is not in mixedCase
Variable MOVIBIT.WBNB (contracts/Token.sol#229) is not in mixedCase
Variable MOVIBIT.DEAD (contracts/Token.sol#230) is not in mixedCase
Variable MOVIBIT.ZERO (contracts/Token.sol#231) is not in mixedCase
Constant MOVIBIT._name (contracts/Token.sol#234) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOVIBIT._symbol (contracts/Token.sol#235) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MOVIBIT._decimals (contracts/Token.sol#236) is not in UPPER_CASE_WITH_UNDERSCORES
Variable MOVIBIT._totalSupply (contracts/Token.sol#238) is not in mixedCase
Variable MOVIBIT._balances (contracts/Token.sol#240) is not in mixedCase
Variable MOVIBIT._allowances (contracts/Token.sol#241) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IDEXRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#182) is too similar to IDEXRouter.addLiquidity(add
ress,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

Auth.authorizationsCount (contracts/Token.sol#120) is never used in MOVIBIT (contracts/Token.sol#226-630)
MOVIBIT.DEAD (contracts/Token.sol#230) is never used in MOVIBIT (contracts/Token.sol#226-630)
MOVIBIT.ZERO (contracts/Token.sol#231) is never used in MOVIBIT (contracts/Token.sol#226-630)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

Auth.authorizationsCount (contracts/Token.sol#120) should be constant
MOVIBIT.DEAD (contracts/Token.sol#230) should be constant
MOVIBIT.WBNB (contracts/Token.sol#229) should be constant
MOVIBIT.ZERO (contracts/Token.sol#231) should be constant
MOVIBIT.routerAddress (contracts/Token.sol#232) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

Auth.creator (contracts/Token.sol#118) should be immutable
MOVIBIT.autoLiquidityReceiver (contracts/Token.sol#264) should be immutable
MOVIBIT.marketingFeeReceiver (contracts/Token.sol#265) should be immutable
MOVIBIT.pair (contracts/Token.sol#268) should be immutable
MOVIBIT.router (contracts/Token.sol#267) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output

FUNCTIONAL TESTING

Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All the functionalities have been tested, no issues were found

1- Adding liquidity (passed):

https://testnet.bscscan.com/tx/0xf9f426ee1f2e9e217fc953c5d64523d8d2c8b1650e32e4af7670d3779c2eeafd

2- Buying when excluded (0% tax) (passed):

https://testnet.bscscan.com/tx/0x719e5c181e4785cda129f88b11f091c8c29e8a44457678cff6ad193918961970

3- Selling when excluded (0% tax) (passed):

https://testnet.bscscan.com/tx/0x3c7ba17db0c4135385d1567cfdda8c95689681c6fec73d4a16ac8636d0333ba2

4- Transferring when excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0xbab4b5d8904f0b34eb95478fa681cabdf425f703c7f85fac522e70fb4a6b7e58

5- Buying when not excluded from fees (up to 15% tax) (passed):

https://testnet.bscscan.com/tx/0x2a663adf4ad014f9e66ecffa56317af02880aadac05003c233bf3467c6fe0b7c

6- Selling when not excluded from fees (up to 15% tax) (passed):

https://testnet.bscscan.com/tx/0x3acd2f0272b1085b3a1e47b8be62f0a47f43eebbb1244599cd685483bc41c3d5

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (up to 15% tax)(passed):
https://testnet.bscscan.com/tx/0x8a98c2b020cb43e24304f379ec94baf0b22515800ad8cd7e37587d1ad54df981

7- Internal swap (burning – auto liquidity – marketing fee) (passed):
https://testnet.bscscan.com/tx/0x3acd2f0272b1085b3a1e47b8be62f0a47f43eebbb1244599cd685483bc41c3d5

MANUAL TESTING

Logical – 0 swap threshold can disable trades

Severity: Critical

function: setSwapBackSettings

Status: Not Resolved

Overview:

if swapThreshold is set to 0, internal swap would be failed due to a division by zero error. (at swapBack function)

```
function setSwapBackSettings(
    bool _enabled,
    uint256 _amount
) external authorized {
    swapEnabled = _enabled;
    swapThreshold = _amount;
}
```

Suggestion

To mitigate this Logical issue, make sure that swapTokensAtAmount is always greater than 0

```
function setSwapBackSettings(
    bool _enabled,
    uint256 _amount
) external authorized {
    swapEnabled = _enabled;
    swapThreshold = _amount;
    require(swapThreshold > totalSupply / 1000000);
}
```

MANUAL TESTING

Centralization – LP tokens going to an EOA

Severity: Medium

function: swapBack

Status: Not Resolved

Overview:

LP tokens generated from auto-liqudity are sent to an EOA account, this accumulated tokens may be used by a malicious owner to remove a portion of tokens from the pool

```
if (amountToLiquify > 0) {
    router.addLiquidityETH{value: amountBNBLiquidity}(
        address(this),
        amountToLiquify,
        0,
        0,
        autoLiquidityReceiver,
        block.timestamp
    );
    emit AutoLiquify(amountBNBLiquidity, amountToLiquify);
}
```

Suggestion

To mitigate this Centralization issue:

- Burn the new LP tokens

or

- Lock the new LP tokens

MANUAL TESTING

Logical – Outdated compiler version

Severity: Low
Status: not resolved
Overview:
Compiler version is 0.7.4, this compiler version is outdated and doesn't support internal handling of overflow/underflows any may introduce other bugs

Recommendation:
Change the compiler version to >= 0.8.0

MANUAL TESTING

Centralization – Excessive fees

Severity: Low
Status: not resolved
Overview:
Fees can be up to 15% for each type of tax (buy/sell/transfer). This is 5% more than what is declared in pinksale safu criteria.

```
function setFeesExchange(
    uint256 _burnFeeBuy,
    uint256 _liquidityFeeBuy,
    uint256 _marketingFeeBuy,
    uint256 _burnFeeSell,
    uint256 _liquidityFeeSell,
    uint256 _marketingFeeSell,
    uint256 _feeDenominator
) external authorized {
    burnFeeBuy = _burnFeeBuy;
    liquidityFeeBuy = _liquidityFeeBuy;
    marketingFeeBuy = _marketingFeeBuy;
    totalFeeBuy = _burnFeeBuy.add(_liquidityFeeBuy).add(_marketingFeeBuy);

    burnFeeSell = _burnFeeSell;
    liquidityFeeSell = _liquidityFeeSell;
    marketingFeeSell = _marketingFeeSell;
    totalFeeSell = _burnFeeSell.add(_liquidityFeeSell).add(
        _marketingFeeSell
    );

    feeDenominator = _feeDenominator;
    require(
        totalFeeBuy <= feeDenominator.div(100).mul(15),
        "Fees cannot be more than 15%"
    );
    require(
        totalFeeSell <= feeDenominator.div(100).mul(15),
        "Fees cannot be more than 15%"
    );
}
```

Recommendation:
Ensure that sum of total buy and total sell fees is less than 25%

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