auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

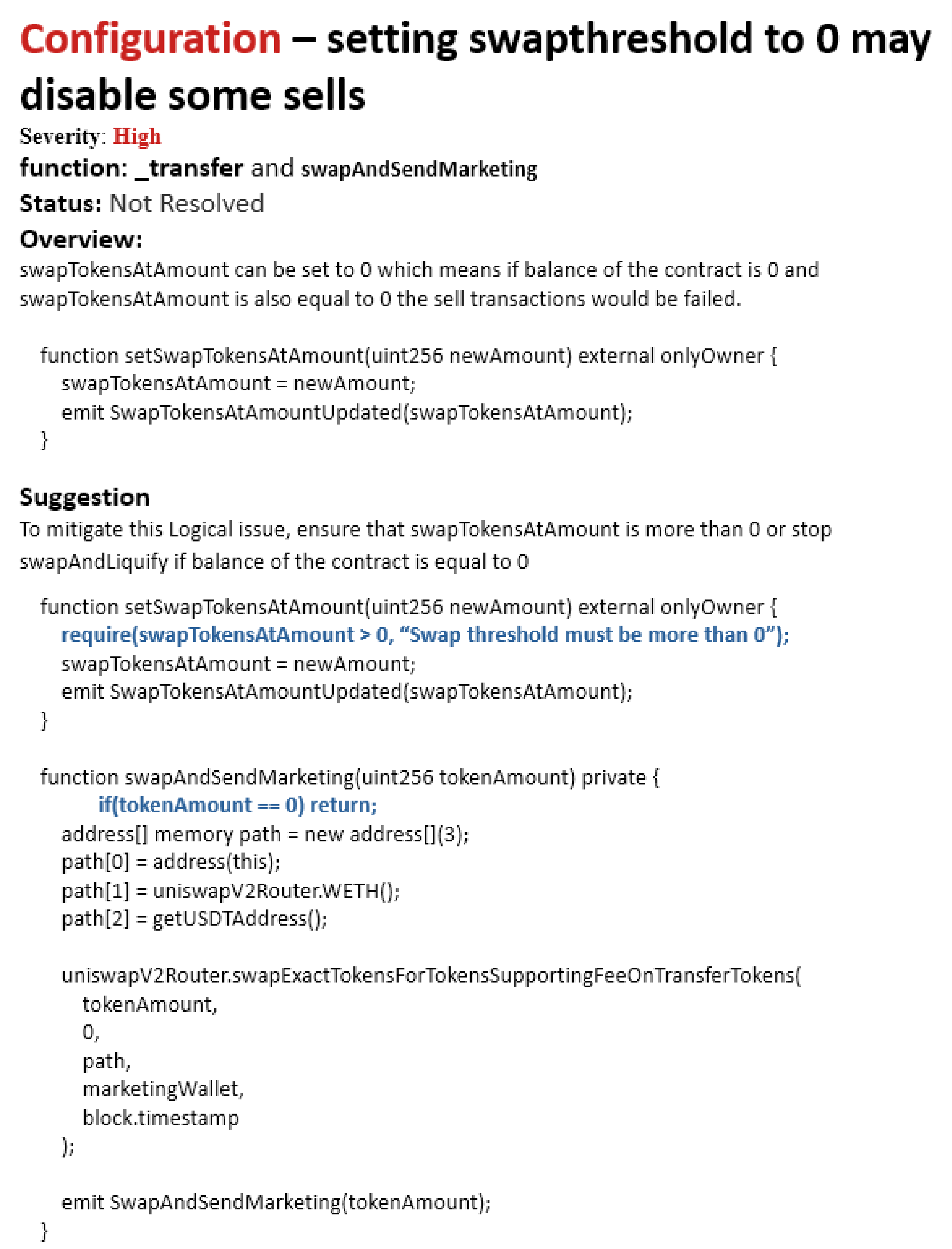
Smart Contract Audit

FOR

ROBINU

DATED : 3 June 23'

HIGH RISK FINDING



Configuration – setting swapthreshold to 0 may disable some sells

Severity: High

function: _transfer and **swapAndSendMarketing**

Status: Not Resolved

Overview:

swapTokensAtAmount can be set to 0 which means if balance of the contract is 0 and swapTokensAtAmount is also equal to 0 the sell transactions would be failed.

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}
```

Suggestion

To mitigate this Logical issue, ensure that swapTokensAtAmount is more than 0 or stop swapAndLiquify if balance of the contract is equal to 0

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    require(swapTokensAtAmount > 0, “Swap threshold must be more than 0”);
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}

function swapAndSendMarketing(uint256 tokenAmount) private {
        if(tokenAmount == 0) return;
    address[] memory path = new address[](3);
    path[0] = address(this);
    path[1] = uniswapV2Router.WETH();
    path[2] = getUSDTAddress();

    uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(
        tokenAmount,
        0,
        path,
        marketingWallet,
        block.timestamp
    );

    emit SwapAndSendMarketing(tokenAmount);
}
```

AUDIT SUMMARY

Project name – ROBINU

Date: 3 June, 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	0	1	0	0	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:
Contract has been tested on binance smart chain testnet which can be found in below link:
https://testnet.bscscan.com/token/0x959CEe6Fc2795A7A7499a02318cBc65773d5b784

Token Information

Token Name : Robinu

Token Symbol: ROBINU

Decimals: 18

Token Supply: 420,690,000,000,000

Token Address: ---

Checksum:
fc26fce3129e55d9f1dcd6e6d5f29de96c7b0c19

Owner:
---**(at time of writing the audit)**

Deployer:

TOKEN OVERVIEW

Fees:

Buy Fees: 0%

Sell Fees: 0%

Transfer Fees: 0%

Fees Privilege: No fees

Ownership: Not Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: - Initial distribution of the tokens

- changing swap threshold

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

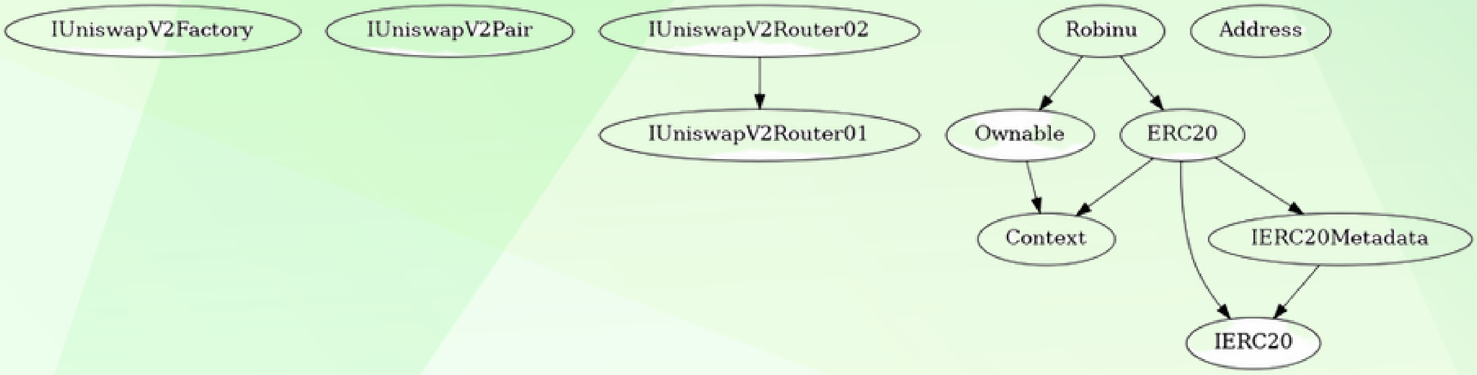
CLASSIFICATION OF RISK

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

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to change buy/sell fees (static 0% fees)
- 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 to 0
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

```
| Contract | Type | Bases | | |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
| └ | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
||||||
| **IUniswapV2Factory** | Interface | |||
| └ | feeTo | External ❗ | |NO ❗ |
| └ | feeToSetter | External ❗ | |NO ❗ |
| └ | getPair | External ❗ | |NO ❗ |
| └ | allPairs | External ❗ | |NO ❗ |
| └ | allPairsLength | External ❗ | |NO ❗ |
| └ | createPair | External ❗ | 🛑 |NO ❗ |
| └ | setFeeTo | External ❗ | 🛑 |NO ❗ |
| └ | setFeeToSetter | External ❗ | 🛑 |NO ❗ |
||||||
| **IUniswapV2Pair** | Interface | |||
| └ | name | External ❗ | |NO ❗ |
| └ | symbol | External ❗ | |NO ❗ |
| └ | decimals | External ❗ | |NO ❗ |
| └ | totalSupply | External ❗ | |NO ❗ |
| └ | balanceOf | External ❗ | |NO ❗ |
| └ | allowance | External ❗ | |NO ❗ |
| └ | approve | External ❗ | 🛑 |NO ❗ |
| └ | transfer | External ❗ | 🛑 |NO ❗ |
| └ | transferFrom | External ❗ | 🛑 |NO ❗ |
| └ | DOMAIN_SEPARATOR | External ❗ | |NO ❗ |
| └ | PERMIT_TYPEHASH | External ❗ | |NO ❗ |
| └ | nonces | External ❗ | |NO ❗ |
| └ | permit | External ❗ | 🛑 |NO ❗ |
| └ | MINIMUM_LIQUIDITY | External ❗ | |NO ❗ |
| └ | factory | External ❗ | |NO ❗ |
| └ | token0 | External ❗ | |NO ❗ |
| └ | token1 | External ❗ | |NO ❗ |
| └ | getReserves | External ❗ | |NO ❗ |
| └ | price0CumulativeLast | External ❗ | |NO ❗ |
| └ | price1CumulativeLast | External ❗ | |NO ❗ |
| └ | kLast | External ❗ | |NO ❗ |
| └ | mint | External ❗ | 🛑 |NO ❗ |
| └ | burn | External ❗ | 🛑 |NO ❗ |
| └ | swap | External ❗ | 🛑 |NO ❗ |
| └ | skim | External ❗ | 🛑 |NO ❗ |
| └ | sync | External ❗ | 🛑 |NO ❗ |
| └ | initialize | External ❗ | 🛑 |NO ❗ |
```

CONTRACT ASSESMENT

```
||||||
| **IUniswapV2Router01** | Interface |  |||
| └ | factory | External ❗ |   |NO ❗ |
| └ | WETH | External ❗ |   |NO ❗ |
| └ | addLiquidity | External ❗ | 🛑  |NO ❗ |
| └ | addLiquidityETH | External ❗ |  💵 |NO ❗ |
| └ | removeLiquidity | External ❗ | 🛑  |NO ❗ |
| └ | removeLiquidityETH | External ❗ | 🛑  |NO ❗ |
| └ | removeLiquidityWithPermit | External ❗ | 🛑  |NO ❗ |
| └ | removeLiquidityETHWithPermit | External ❗ | 🛑  |NO ❗ |
| └ | swapExactTokensForTokens | External ❗ | 🛑  |NO ❗ |
| └ | swapTokensForExactTokens | External ❗ | 🛑  |NO ❗ |
| └ | swapExactETHForTokens | External ❗ |  💵 |NO ❗ |
| └ | swapTokensForExactETH | External ❗ | 🛑  |NO ❗ |
| └ | swapExactTokensForETH | External ❗ | 🛑  |NO ❗ |
| └ | swapETHForExactTokens | External ❗ |  💵 |NO ❗ |
| └ | quote | External ❗ |   |NO ❗ |
| └ | getAmountOut | External ❗ |   |NO ❗ |
| └ | getAmountIn | External ❗ |   |NO ❗ |
| └ | getAmountsOut | External ❗ |   |NO ❗ |
| └ | getAmountsIn | External ❗ |   |NO ❗ |
||||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
| └ | removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO ❗ |
| └ | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO ❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO ❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ |  💵 |NO ❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑  |NO ❗ |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗ |   |NO ❗ |
| └ | balanceOf | External ❗ |   |NO ❗ |
| └ | transfer | External ❗ | 🛑  |NO ❗ |
| └ | allowance | External ❗ |   |NO ❗ |
| └ | approve | External ❗ | 🛑  |NO ❗ |
| └ | transferFrom | External ❗ | 🛑  |NO ❗ |
||||||
| **IERC20Metadata** | Interface | IERC20 |||
| └ | name | External ❗ |   |NO ❗ |
| └ | symbol | External ❗ |   |NO ❗ |
| └ | decimals | External ❗ |   |NO ❗ |
||||||
```

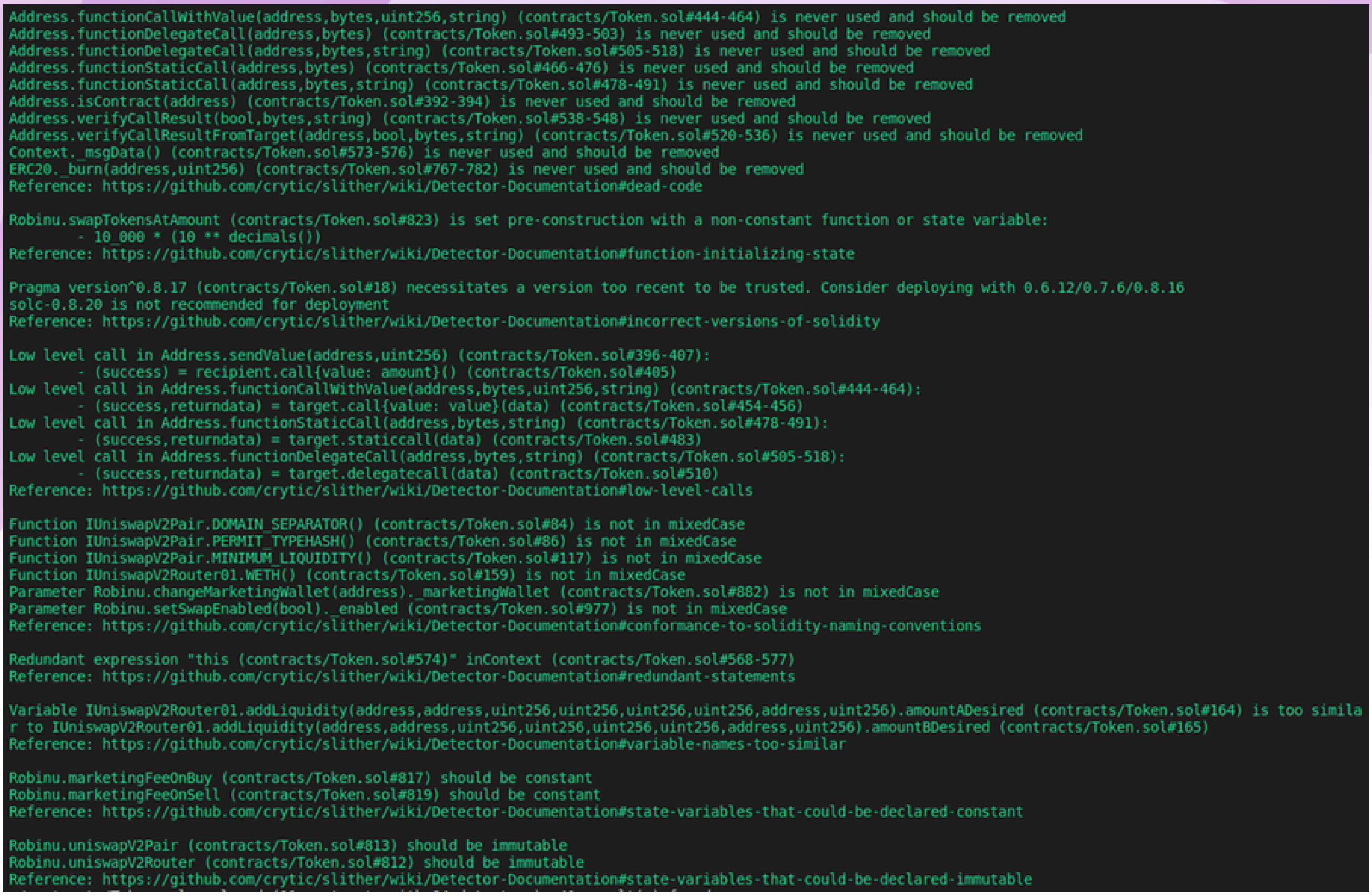
CONTRACT ASSESMENT

```
| **Address** | Library |  |||
| └ | isContract | Internal 🔒 |   | |
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | verifyCallResultFromTarget | Internal 🔒 |   | |
| └ | verifyCallResult | Internal 🔒 |   | |
| └ | _revert | Private 🔐 |   | |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | owner | Public ❗ |   |NO❗ |
| └ | renounceOwnership | Public ❗ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗ | 🛑  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑  | |
||||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | name | Public ❗ |   |NO❗ |
| └ | symbol | Public ❗ |   |NO❗ |
| └ | decimals | Public ❗ |   |NO❗ |
| └ | totalSupply | Public ❗ |   |NO❗ |
| └ | balanceOf | Public ❗ |   |NO❗ |
| └ | transfer | Public ❗ | 🛑  |NO❗ |
| └ | allowance | Public ❗ |   |NO❗ |
| └ | approve | Public ❗ | 🛑  |NO❗ |
| └ | transferFrom | Public ❗ | 🛑  |NO❗ |
| └ | increaseAllowance | Public ❗ | 🛑  |NO❗ |
| └ | decreaseAllowance | Public ❗ | 🛑  |NO❗ |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑 | |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑 | |
||||||
| **Robinu** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗️ | 🛑 | ERC20 |
| └ | <Receive Ether> | External ❗️ | 💵 |NO❗️ |
| └ | getUSDTAddress | Public ❗️ | |NO❗️ |
| └ | isContract | Internal 🔒 | | |
| └ | changeMarketingWallet | External ❗️ | 🛑 | onlyOwner |
| └ | claimStuckTokens | External ❗️ | 🛑 | onlyOwner |
| └ | excludeFromFees | External ❗️ | 🛑 | onlyOwner |
| └ | isExcludedFromFees | Public ❗️ | |NO❗️ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | setSwapEnabled | External ❗️ | 🛑 | onlyOwner |
| └ | setSwapTokensAtAmount | External ❗️ | 🛑 | onlyOwner |
| └ | swapAndSendMarketing | Private 🔐 | 🛑 | |


### Legend

| Symbol | Meaning |
|:--------:|-----------|
| 🛑 | Function can modify state |
| 💵 | Function is payable |
```

STATIC ANALYSIS



```
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#444-464) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/Token.sol#493-503) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#505-518) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/Token.sol#466-476) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#478-491) is never used and should be removed
Address.isContract(address) (contracts/Token.sol#392-394) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/Token.sol#538-548) is never used and should be removed
Address.verifyCallResultFromTarget(address,bool,bytes,string) (contracts/Token.sol#520-536) is never used and should be removed
Context._msgData() (contracts/Token.sol#573-576) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#767-782) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Robinu.swapTokensAtAmount (contracts/Token.sol#823) is set pre-construction with a non-constant function or state variable:
        - 10_000 * (10 ** decimals())
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#18) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.20 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#396-407):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#405)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#444-464):
        - (success,returndata) = target.call{value: value}(data) (contracts/Token.sol#454-456)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#478-491):
        - (success,returndata) = target.staticcall(data) (contracts/Token.sol#483)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#505-518):
        - (success,returndata) = target.delegatecall(data) (contracts/Token.sol#510)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#84) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#86) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#117) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#159) is not in mixedCase
Parameter Robinu.changeMarketingWallet(address)._marketingWallet (contracts/Token.sol#882) is not in mixedCase
Parameter Robinu.setSwapEnabled(bool)._enabled (contracts/Token.sol#977) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#574)" inContext (contracts/Token.sol#568-577)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#164) is too simila
r to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#165)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

Robinu.marketingFeeOnBuy (contracts/Token.sol#817) should be constant
Robinu.marketingFeeOnSell (contracts/Token.sol#819) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

Robinu.uniswapV2Pair (contracts/Token.sol#813) should be immutable
Robinu.uniswapV2Router (contracts/Token.sol#812) 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

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x926b9c6b912b9fad052db21c310a55a0fdd0091c40eefd7339675db6d62874d2

2- Buying when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x07589f87cd16fa20c726db67c308363c8285e1c124e0c319f7e230594dedaab0

3- Selling when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xfda9a933be6a9ed31aa4ea41dbe9ae38282aae69211b4a4c9b726a3e09ec3d31

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x0e06477d6b62373fe2895f8bb04bdf86754e570ba35877272d82c409e6410d55

5- Buying when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xb554be130fae48e61902da28272574ae77ffe430534ddd905ef488eae07fff1c

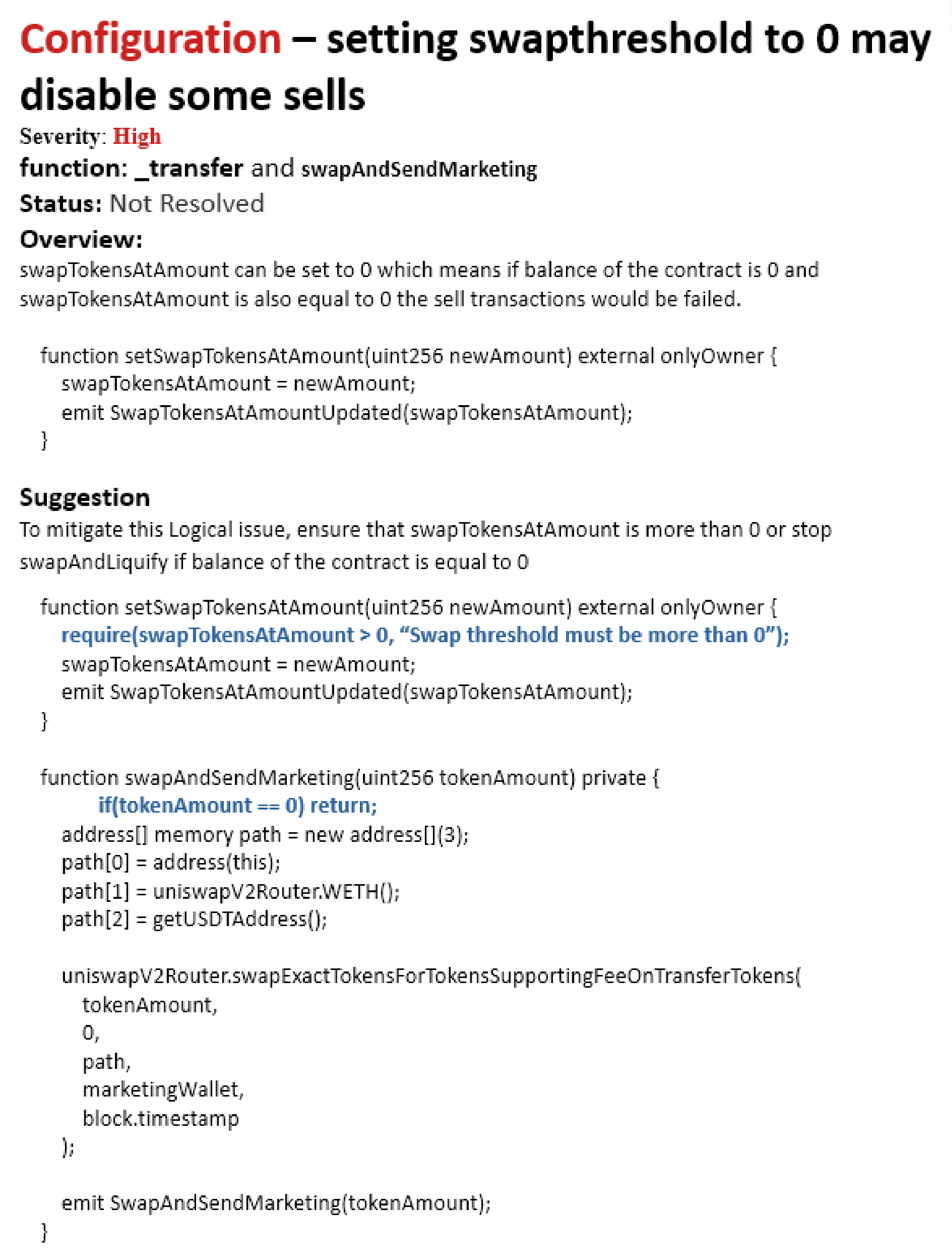
6- Selling when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x6d0b4bae607df5463ecd99161ad1e84012c3f9391906fd3c775880586a435830

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0x41183a26ce57658fefc58836a3fb84b184b1b9a9b1185e34829aecaffe8de8e6

FUNCTIONAL TESTING



Configuration – setting swapthreshold to 0 may disable some sells

Severity: High

function: _transfer and **swapAndSendMarketing**

Status: Not Resolved

Overview:

swapTokensAtAmount can be set to 0 which means if balance of the contract is 0 and swapTokensAtAmount is also equal to 0 the sell transactions would be failed.

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}
```

Suggestion

To mitigate this Logical issue, ensure that swapTokensAtAmount is more than 0 or stop swapAndLiquify if balance of the contract is equal to 0

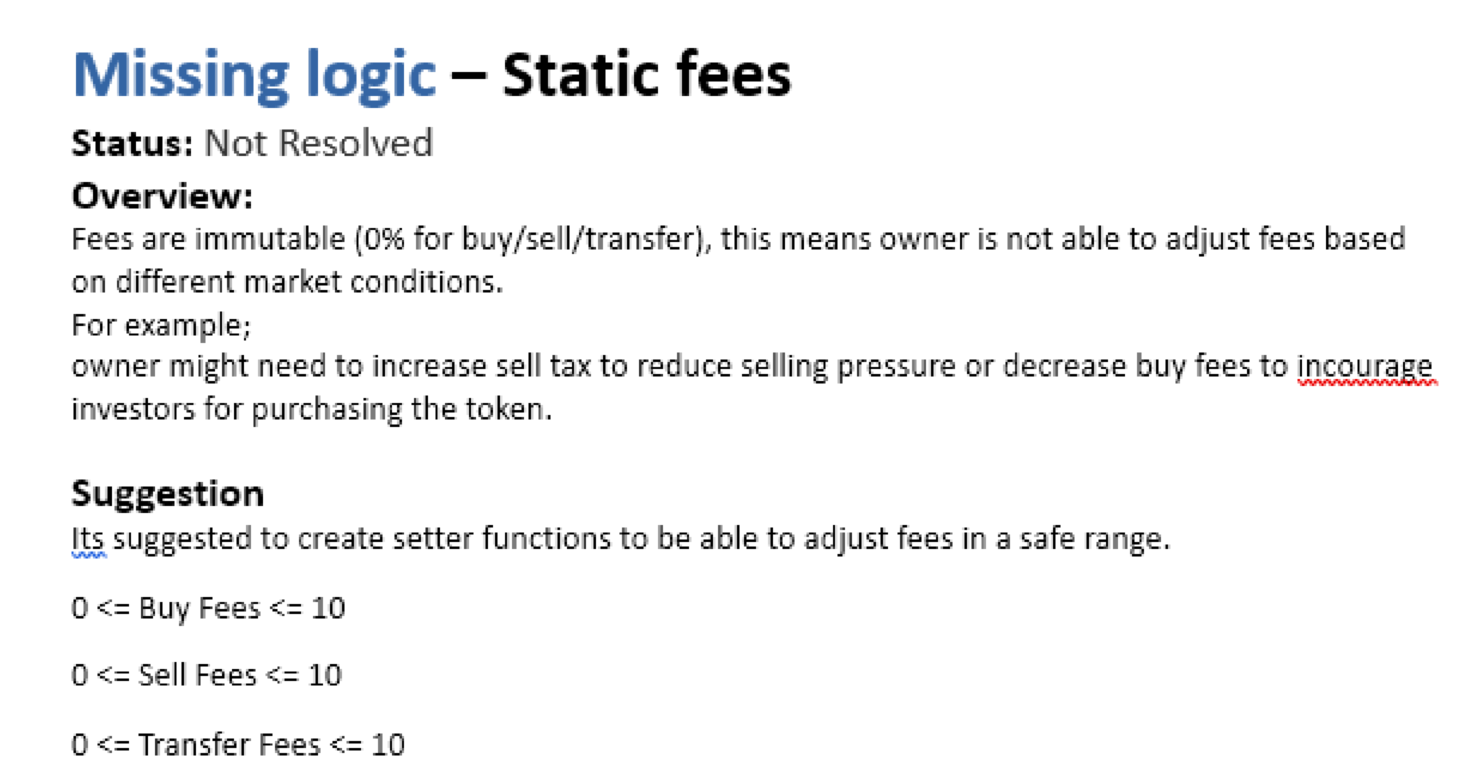
```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    require(swapTokensAtAmount > 0, “Swap threshold must be more than 0”);
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}

function swapAndSendMarketing(uint256 tokenAmount) private {
        if(tokenAmount == 0) return;
    address[] memory path = new address[](3);
    path[0] = address(this);
    path[1] = uniswapV2Router.WETH();
    path[2] = getUSDTAddress();

    uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(
        tokenAmount,
        0,
        path,
        marketingWallet,
        block.timestamp
    );

    emit SwapAndSendMarketing(tokenAmount);
}
```

FUNCTIONAL TESTING



Missing logic – Static fees

Status: Not Resolved

Overview:
Fees are immutable (0% for buy/sell/transfer), this means owner is not able to adjust fees based on different market conditions.
For example;
owner might need to increase sell tax to reduce selling pressure or decrease buy fees to incourage investors for purchasing the token.

Suggestion
Its suggested to create setter functions to be able to adjust fees in a safe range.

0 <= Buy Fees <= 10

0 <= Sell Fees <= 10

0 <= Transfer Fees <= 10

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