auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

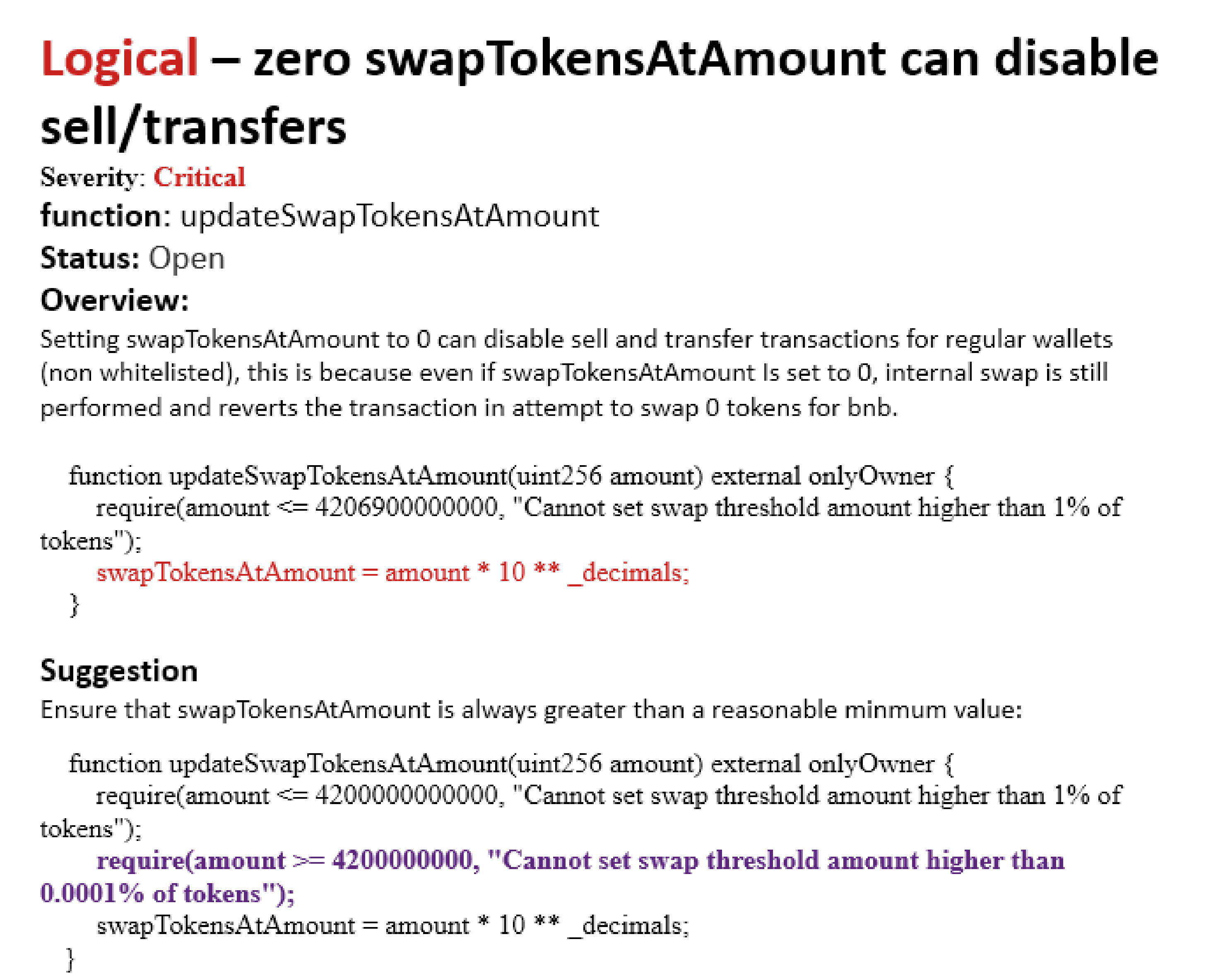
Smart Contract Audit

FOR

BOSS

DATED : 5 June 23'

CRITICAL RISK FINDING



Logical – zero swapTokensAtAmount can disable sell/transfers

Severity: Critical

function: updateSwapTokensAtAmount

Status: Open

Overview:

Setting swapTokensAtAmount to 0 can disable sell and transfer transactions for regular wallets (non whitelisted), this is because even if swapTokensAtAmount Is set to 0, internal swap is still performed and reverts the transaction in attempt to swap 0 tokens for bnb.

```
    function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
        require(amount <= 4206900000000, "Cannot set swap threshold amount higher than 1% of
tokens");
        swapTokensAtAmount = amount * 10 ** _decimals;
    }
```

Suggestion

Ensure that swapTokensAtAmount is always greater than a reasonable minmum value:

```
    function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
        require(amount <= 4200000000000, "Cannot set swap threshold amount higher than 1% of
tokens");
        require(amount >= 4200000000, "Cannot set swap threshold amount higher than
0.0001% of tokens");
        swapTokensAtAmount = amount * 10 ** _decimals;
    }
```

HIGH RISK FINDING

Centralization – Trades must be enabled

Severity: **High**

function: EnableTrading

Status: Open

Overview:

The smart contract owner must enable trades for holders. If trading remain disabled, no one would be able to buy/sell/transfer tokens.

```
function EnableTrading() external onlyOwner {
    require(!tradingEnabled, "Cannot re-enable trading");
    tradingEnabled = true;
    swapEnabled = true;
    genesis_block = block.number;
}
```

Suggestion

To mitigate this centralization issue, we propose the following options:

1. Renounce Ownership: Consider relinquishing control of the smart contract by renouncing ownership. This would remove the ability for a single entity to manipulate the router, reducing centralization risks.
2. Multi-signature Wallet: Transfer ownership to a multi-signature wallet. This would require multiple approvals for any changes to the mainRouter, adding an additional layer of security and reducing the centralization risk.
3. Transfer ownership to a trusted and valid 3rd party in order to guarantee enabling of the trades

AUDIT SUMMARY

Project name – BOSS

Date: 5 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	1	0	1	0	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/0x47D63190157F1ade0Ea353b7ec08c0acA554808B

Token Information

Token Name : Boss Baby Inu

Token Symbol: BOSS

Decimals: 9

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

Token Address:
0xA023CaCaf2f4eF473705b6c1172509EF2343dc84

Checksum:
117db274a688d31b79a4d6834d3d691b7c3b2f52

Owner:
0x142523Ab1D9199BD9eb0a0d5f7865992321C04Dc
(at time of writing the audit)

Deployer:
0x142523Ab1D9199BD9eb0a0d5f7865992321C04Dc

TOKEN OVERVIEW

Fees:

Buy Fees: 8%

Sell Fees: 8%

Transfer Fees: 8%

Fees Privilege: None

Ownership: Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: - changing swap threshold

- enabling trades

- 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 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

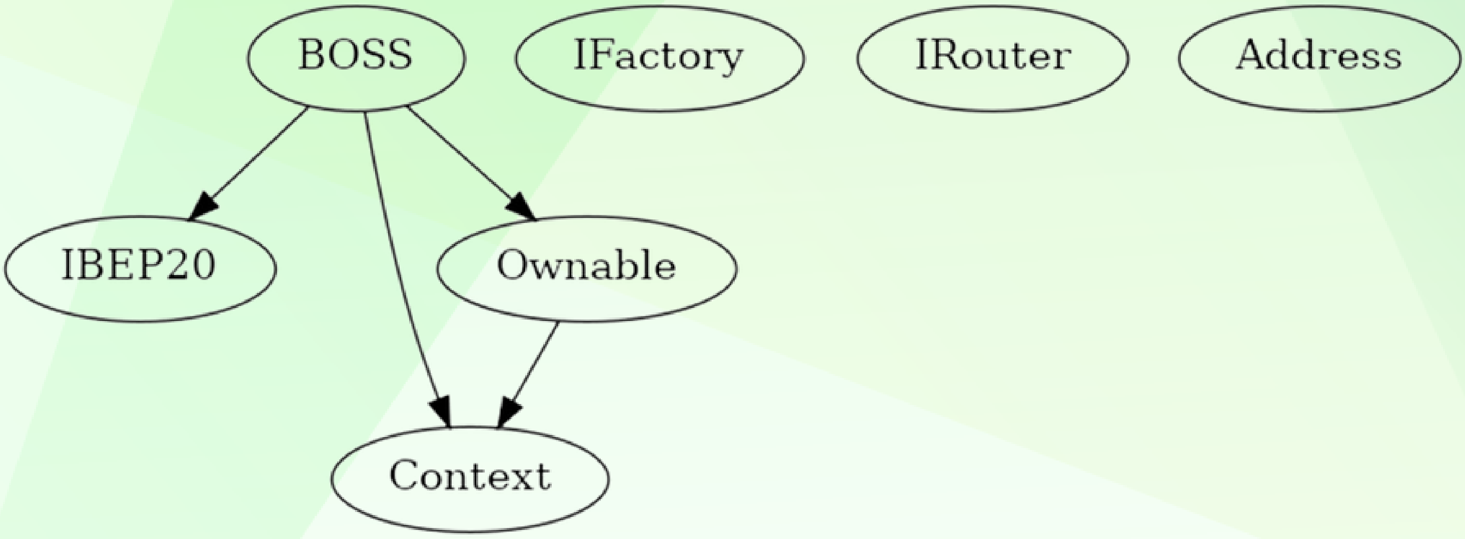
CLASSIFICATION OF RISK

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	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to change fees (8% for each type of tax)
- 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
- Owner must enable trades manually

CONTRACT ASSESMENT

```
|  Contract  |        Type        |       Bases      |                  |                 |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |  **Modifiers**  |
||||||
| **IBEP20** | Interface |  |||
| └ | totalSupply | External ❗  |   |NO ❗  |
| └ | balanceOf | External ❗  |   |NO ❗  |
| └ | transfer | External ❗  | 🛑  |NO ❗  |
| └ | allowance | External ❗  |   |NO ❗  |
| └ | approve | External ❗  | 🛑  |NO ❗  |
| └ | transferFrom | External ❗  | 🛑  |NO ❗  |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒  |   | |
| └ | _msgData | Internal 🔒  |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗  | 🛑  |NO ❗  |
| └ | owner | Public ❗  |   |NO ❗  |
| └ | renounceOwnership | Public ❗  | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗  | 🛑  | onlyOwner |
| └ | _setOwner | Private 🔐  | 🛑  | |
||||||
| **IFactory** | Interface |  |||
| └ | createPair | External ❗  | 🛑  |NO ❗  |
||||||
| **IRouter** | Interface |  |||
| └ | factory | External ❗  |   |NO ❗  |
| └ | WETH | External ❗  |   |NO ❗  |
| └ | addLiquidityETH | External ❗  | 💵  |NO ❗  |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗  | 🛑  |NO ❗  |
||||||
| **Address** | Library |  |||
| └ | sendValue | Internal 🔒  | 🛑  | |
||||||
| **BOSS** | Implementation | Context, IBEP20, Ownable |||
| └ | <Constructor> | Public ❗  | 🛑  |NO ❗  |
| └ | name | Public ❗  |   |NO ❗  |
| └ | symbol | Public ❗  |   |NO ❗  |
| └ | decimals | Public ❗  |   |NO ❗  |
| └ | totalSupply | Public ❗  |   |NO ❗  |
| └ | balanceOf | Public ❗  |   |NO ❗  |
| └ | allowance | Public ❗  |   |NO ❗  |
| └ | approve | Public ❗  | 🛑  |NO ❗  |
```

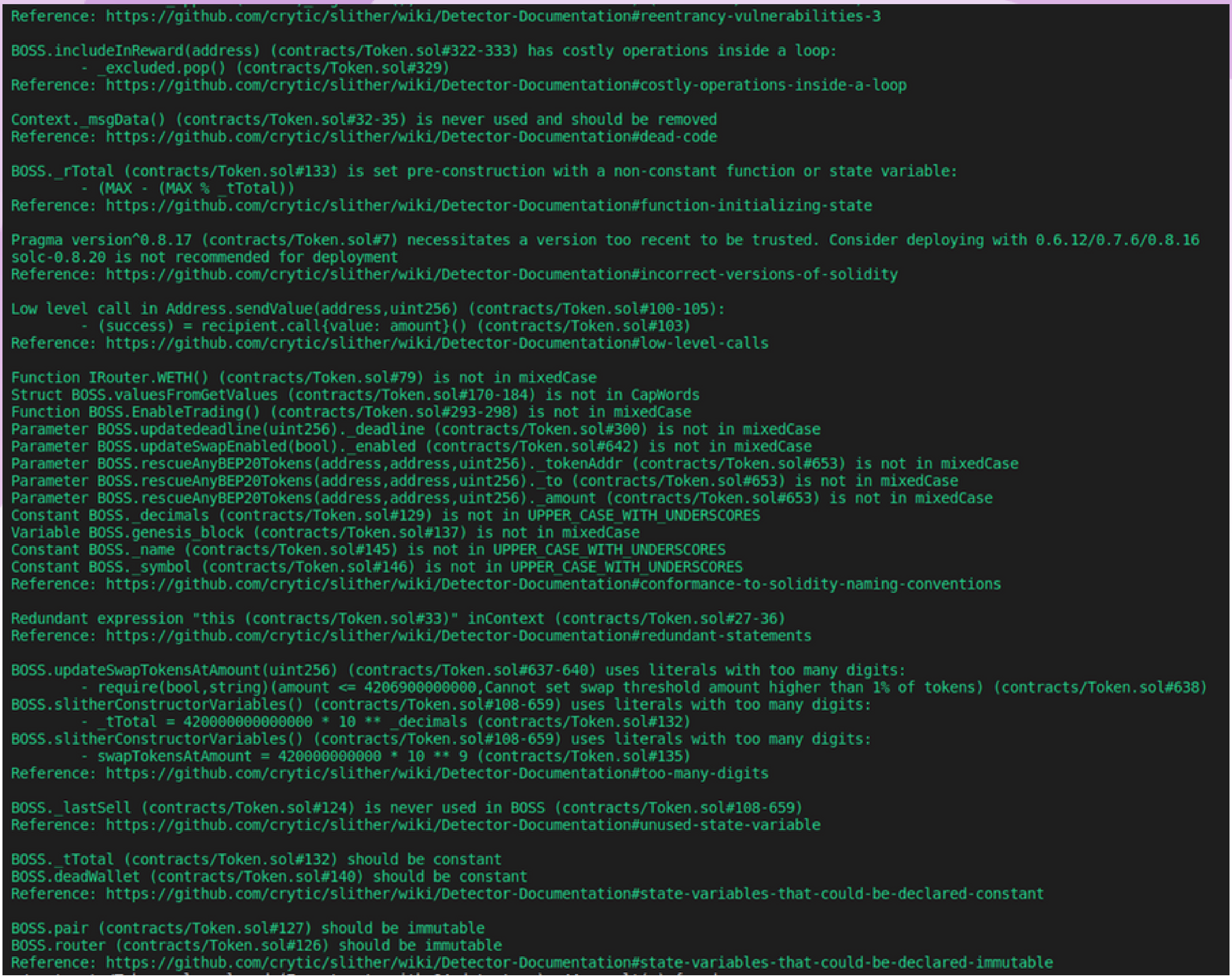
CONTRACT ASSESMENT

```
| └ | transferFrom | Public ❗️ | 🛑  |NO ❗️ |
| └ | increaseAllowance | Public ❗️ | 🛑  |NO ❗️ |
| └ | decreaseAllowance | Public ❗️ | 🛑  |NO ❗️ |
| └ | transfer | Public ❗️ | 🛑  |NO ❗️ |
| └ | isExcludedFromReward | Public ❗️ |   |NO ❗️ |
| └ | reflectionFromToken | Public ❗️ |   |NO ❗️ |
| └ | EnableTrading | External ❗️ | 🛑  | onlyOwner |
| └ | updatedeadline | External ❗️ | 🛑  | onlyOwner |
| └ | tokenFromReflection | Public ❗️ |   |NO ❗️ |
| └ | excludeFromReward | Public ❗️ | 🛑  | onlyOwner |
| └ | includeInReward | External ❗️ | 🛑  | onlyOwner |
| └ | excludeFromFee | Public ❗️ | 🛑  | onlyOwner |
| └ | includeInFee | Public ❗️ | 🛑  | onlyOwner |
| └ | isExcludedFromFee | Public ❗️ |   |NO ❗️ |
| └ | _reflectRfi | Private 🔐 | 🛑  | |
| └ | _takeLiquidity | Private 🔐 | 🛑  | |
| └ | _takeMarketing | Private 🔐 | 🛑  | |
| └ | _takeOps | Private 🔐 | 🛑  | |
| └ | _takeDev | Private 🔐 | 🛑  | |
| └ | _getValues | Private 🔐 |   | |
| └ | _getTValues | Private 🔐 |   | |
| └ | _getRValues1 | Private 🔐 |   | |
| └ | _getRValues2 | Private 🔐 |   | |
| └ | _getRate | Private 🔐 |   | |
| └ | _getCurrentSupply | Private 🔐 |   | |
| └ | _approve | Private 🔐 | 🛑  | |
| └ | _transfer | Private 🔐 | 🛑  | |
| └ | _tokenTransfer | Private 🔐 | 🛑  | |
| └ | swapAndLiquify | Private 🔐 | 🛑  | lockTheSwap |
| └ | addLiquidity | Private 🔐 | 🛑  | |
| └ | swapTokensForBNB | Private 🔐 | 🛑  | |
| └ | bulkExcludeFromFee | External ❗️ | 🛑  | onlyOwner |
| └ | bulkIncludeInFee | External ❗️ | 🛑  | onlyOwner |
| └ | updateMarketingWallet | External ❗️ | 🛑  | onlyOwner |
| └ | updateDevWallet | External ❗️ | 🛑  | onlyOwner |
| └ | updateOpsWallet | External ❗️ | 🛑  | onlyOwner |
| └ | updateSwapTokensAtAmount | External ❗️ | 🛑  | onlyOwner |
| └ | updateSwapEnabled | External ❗️ | 🛑  | onlyOwner |
| └ | rescueBNB | External ❗️ | 🛑  | onlyOwner |
| └ | rescueAnyBEP20Tokens | Public ❗️ | 🛑  | onlyOwner |
| └ | <Receive Ether> | External ❗️ | 💵 |NO ❗️ |
```

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

BOSS.includeInReward(address) (contracts/Token.sol#322-333) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#329)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#32-35) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

BOSS._rTotal (contracts/Token.sol#133) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#7) 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#100-105):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#103)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#79) is not in mixedCase
Struct BOSS.valuesFromGetValues (contracts/Token.sol#170-184) is not in CapWords
Function BOSS.EnableTrading() (contracts/Token.sol#293-298) is not in mixedCase
Parameter BOSS.updatedeadline(uint256)._deadline (contracts/Token.sol#300) is not in mixedCase
Parameter BOSS.updateSwapEnabled(bool)._enabled (contracts/Token.sol#642) is not in mixedCase
Parameter BOSS.rescueAnyBEP20Tokens(address,address,uint256)._tokenAddr (contracts/Token.sol#653) is not in mixedCase
Parameter BOSS.rescueAnyBEP20Tokens(address,address,uint256)._to (contracts/Token.sol#653) is not in mixedCase
Parameter BOSS.rescueAnyBEP20Tokens(address,address,uint256)._amount (contracts/Token.sol#653) is not in mixedCase
Constant BOSS._decimals (contracts/Token.sol#129) is not in UPPER_CASE_WITH_UNDERSCORES
Variable BOSS.genesis_block (contracts/Token.sol#137) is not in mixedCase
Constant BOSS._name (contracts/Token.sol#145) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BOSS._symbol (contracts/Token.sol#146) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#33)" inContext (contracts/Token.sol#27-36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

BOSS.updateSwapTokensAtAmount(uint256) (contracts/Token.sol#637-640) uses literals with too many digits:
        - require(bool,string)(amount <= 4206900000000,Cannot set swap threshold amount higher than 1% of tokens) (contracts/Token.sol#638)
BOSS.slitherConstructorVariables() (contracts/Token.sol#108-659) uses literals with too many digits:
        - _tTotal = 420000000000000 * 10 ** _decimals (contracts/Token.sol#132)
BOSS.slitherConstructorVariables() (contracts/Token.sol#108-659) uses literals with too many digits:
        - swapTokensAtAmount = 420000000000 * 10 ** 9 (contracts/Token.sol#135)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

BOSS._lastSell (contracts/Token.sol#124) is never used in BOSS (contracts/Token.sol#108-659)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

BOSS._tTotal (contracts/Token.sol#132) should be constant
BOSS.deadWallet (contracts/Token.sol#140) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

BOSS.pair (contracts/Token.sol#127) should be immutable
BOSS.router (contracts/Token.sol#126) 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/0xdbdbec9ebe6d281c64aa07e84f84483c352ba71782e910cacbb9bc98387be50e

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x89f713f7dc21e484b650b9444ae3cb0327c6c4672c4ec62417fa7dbf031fd5a2

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x8ae83d80bccf401293c231138a475ca39c78e7224595f5139af708f6dc400940

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x8dbd99c1e482189507a2fe3cda777af46e81d8bc05b3165377a39fe35d39f5b5

5- Buying when not excluded from fees (8% tax) (passed):
https://testnet.bscscan.com/tx/0x1300c3310090a1903272dd902ea9ae4d435bddddaf1cc030db1debebc08a2e18

6- Selling when not excluded from fees (8% tax) (passed):
https://testnet.bscscan.com/tx/0x2966c984842c11372479c9ef18978bed81c84a6c82057a7dba68939881810c26

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (8% tax) **(passed):**

https://testnet.bscscan.com/tx/0x7604ae58f78dec9e093362f0ff6af0dc3be040d4dcd7ccacd82e7ff34d005812

8- Internal swap (marketing and ops wallets received BNB) **(passed):**

https://testnet.bscscan.com/tx/0x2966c984842c11372479c9ef18978bed81c84a6c82057a7dba68939881810c26

FUNCTIONAL TESTING

Logical – zero swapTokensAtAmount can disable sell/transfers

Severity: Critical

function: updateSwapTokensAtAmount

Status: Open

Overview:

Setting swapTokensAtAmount to 0 can disable sell and transfer transactions for regular wallets (non whitelisted), this is because even if swapTokensAtAmount Is set to 0, internal swap is still performed and reverts the transaction in attempt to swap 0 tokens for bnb.

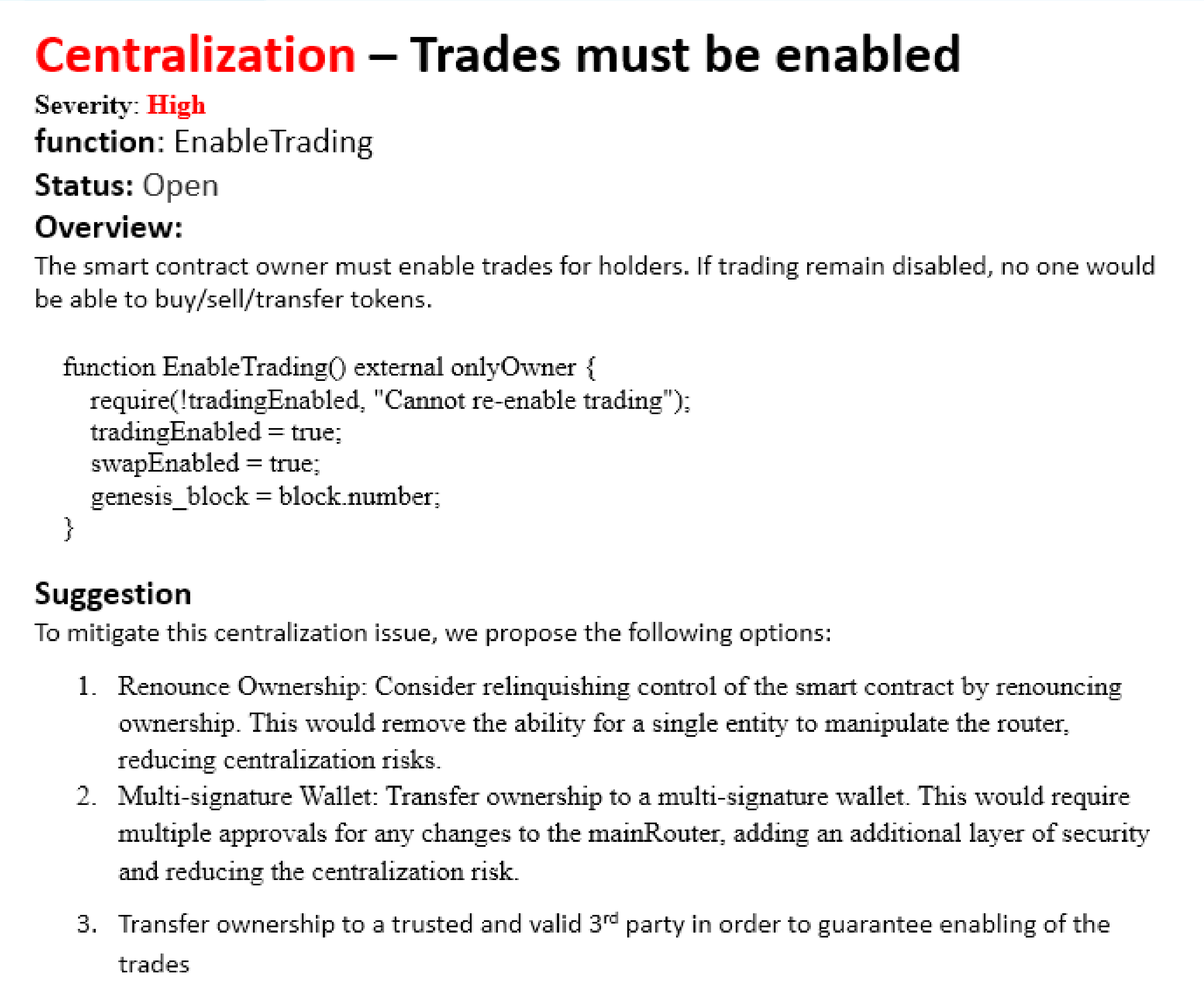
```
  function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
      require(amount <= 4206900000000, "Cannot set swap threshold amount higher than 1% of
tokens");
      swapTokensAtAmount = amount * 10 ** _decimals;
  }
```

Suggestion

Ensure that swapTokensAtAmount is always greater than a reasonable minmum value:

```
  function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
      require(amount <= 4200000000000, "Cannot set swap threshold amount higher than 1% of
tokens");
      require(amount >= 4200000000, "Cannot set swap threshold amount higher than
0.0001% of tokens");
      swapTokensAtAmount = amount * 10 ** _decimals;
  }
```

FUNCTIONAL TESTING



Centralization – Trades must be enabled

Severity: **High**

function: EnableTrading

Status: Open

Overview:

The smart contract owner must enable trades for holders. If trading remain disabled, no one would be able to buy/sell/transfer tokens.

```
function EnableTrading() external onlyOwner {
    require(!tradingEnabled, "Cannot re-enable trading");
    tradingEnabled = true;
    swapEnabled = true;
    genesis_block = block.number;
}
```

Suggestion

To mitigate this centralization issue, we propose the following options:

1. Renounce Ownership: Consider relinquishing control of the smart contract by renouncing ownership. This would remove the ability for a single entity to manipulate the router, reducing centralization risks.
2. Multi-signature Wallet: Transfer ownership to a multi-signature wallet. This would require multiple approvals for any changes to the mainRouter, adding an additional layer of security and reducing the centralization risk.
3. Transfer ownership to a trusted and valid 3rd party in order to guarantee enabling of the trades

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