auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Dominator Domain

DATED : 20 MAR 23'

AUDIT SUMMARY

Project name – Dominator Domain

Date: 20 March, 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	1	0	0	2
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 Testnet network:
all tests were done on Bsc Testnet network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

https://testnet.bscscan.com/address/0xBa590cb3400e1d0355bdaA9D435700b57D3AeB1b#code

Token Information

Token Name : Dominator Domain

Token Symbol: DomDom

Decimals: 18

Token Supply: 500,000,000

Token Address:
0x635d0e13f98e107Cf6C5cDFbF52C19843F87e76a

Checksum:
7539e92eafdc416f8c4d4550dec56f40ed0ef55b

Owner:
0x5E12DE4284670cF35eF793A27042A805B36aE848

Deployer:
0x8134b687be5752eFF8361B663030420D47648bfF

TOKEN OVERVIEW

Fees:

Buy Fees: 0%

Sell Fees: 3%

Transfer Fees: 0%

Fees Privilige: None

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: Including in fee - excluding from fee

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

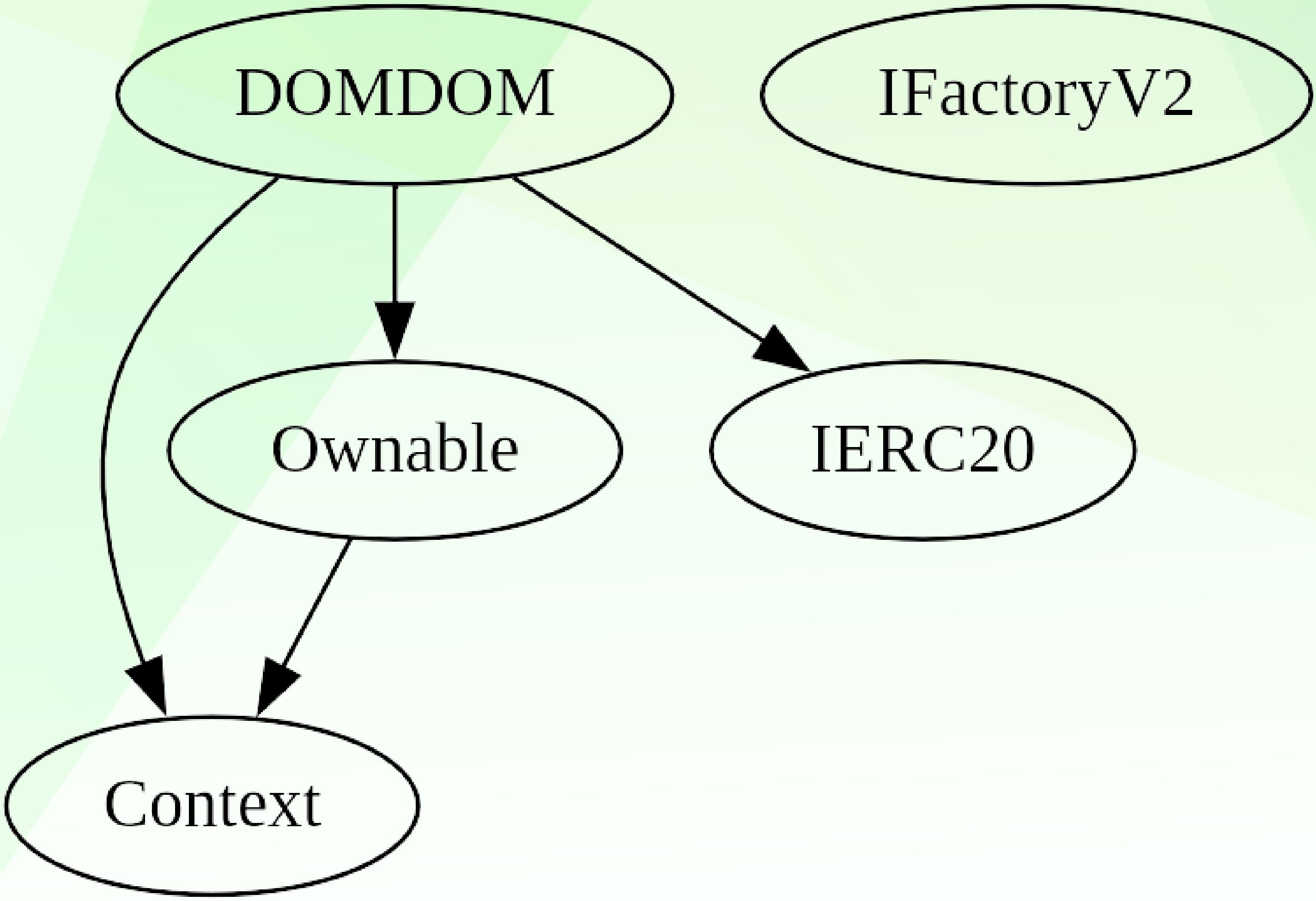
CLASSIFICATION OF RISK

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	2

INHERITANCE TREE



POINTS TO NOTE

- **Owner must enable trading for investors**
- Owner is able to change buy/sell fees up to 3% each one (0% transfer fee)
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens

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❗ |
||||||
| **IRouter02** | Interface | IRouter01 |||
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 |NO❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO❗ |
| └ | swapExactTokensForTokens | External ❗ | 🛑 |NO❗ |
```

CONTRACT ASSESMENT

```
||||||
| **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❗️ |
||||||
| **DOMDOM** | 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❗️ |
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | <Receive Ether> | External ❗️ | 💵 |NO❗️ |
| └ | transfer | Public ❗️ | 🛑  |NO❗️ |
| └ | approve | External ❗️ | 🛑  |NO❗️ |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | transferFrom | External ❗️ | 🛑  |NO❗️ |
| └ | isNoFeeWalelt | External ❗️ |   |NO❗️ |
| └ | setNoFeeWallet | Public ❗️ | 🛑  | onlyOwner |
| └ | isLimitedAddress | Internal 🔒 |   | |
| └ | is_buy | Internal 🔒 |   | |
| └ | is_sell | Internal 🔒 |   | |
| └ | is_transfer | Internal 🔒 |   | |
| └ | canSwap | Internal 🔒 |   | |
| └ | changeLpPair | External ❗️ | 🛑  | onlyOwner |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _basicTransfer | Internal 🔒 | 🛑  | |
| └ | changeWallets | External ❗️ |  💵 | onlyOwner |
| └ | takeTaxes | Internal 🔒 | 🛑  | |
| └ | internalSwap | Internal 🔒 | 🛑  | inSwapFlag |
| └ | updateBuyFeeAmount | External ❗️ | 🛑  | onlyOwner |
```

CONTRACT ASSESMENT

└	updateSellFeeAmount	External ❗	🛑	onlyOwner
└	setPresaleAddress	External ❗	🛑	onlyOwner
└	enableTrading	External ❗	🛑	onlyOwner
└	rescueETH	External ❗	🛑	onlyOwner
└	rescueERC20	External ❗	🛑	onlyOwner

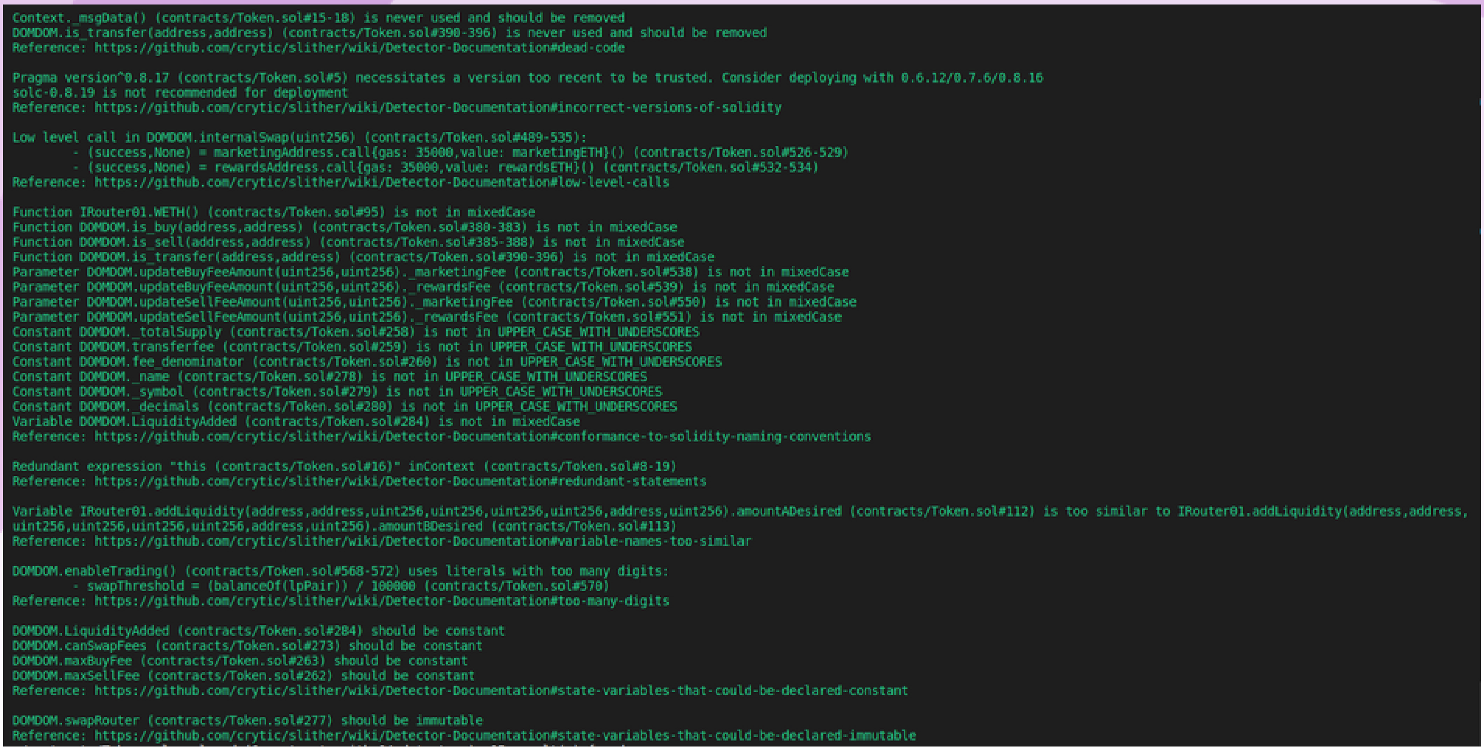
Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

TOKEN DISTRIBUTION

It should be noted that the owner currently holds 100% of the total supply. However, information about the distribution of these tokens is not available, and it is recommended that investors exercise caution when considering this aspect.

STATIC ANALYSIS



```
Context._msgData() (contracts/Token.sol#15-18) is never used and should be removed
DOMDOM.is_transfer(address,address) (contracts/Token.sol#390-396) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.17 (contracts/Token.sol#5) 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 DOMDOM.internalSwap(uint256) (contracts/Token.sol#489-535):
        - (success,None) = marketingAddress.call{gas: 35000,value: marketingETH}() (contracts/Token.sol#526-529)
        - (success,None) = rewardsAddress.call{gas: 35000,value: rewardsETH}() (contracts/Token.sol#532-534)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter01.WETH() (contracts/Token.sol#95) is not in mixedCase
Function DOMDOM.is_buy(address,address) (contracts/Token.sol#380-383) is not in mixedCase
Function DOMDOM.is_sell(address,address) (contracts/Token.sol#385-388) is not in mixedCase
Function DOMDOM.is_transfer(address,address) (contracts/Token.sol#390-396) is not in mixedCase
Parameter DOMDOM.updateBuyFeeAmount(uint256,uint256)._marketingFee (contracts/Token.sol#538) is not in mixedCase
Parameter DOMDOM.updateBuyFeeAmount(uint256,uint256)._rewardsFee (contracts/Token.sol#539) is not in mixedCase
Parameter DOMDOM.updateSellFeeAmount(uint256,uint256)._marketingFee (contracts/Token.sol#550) is not in mixedCase
Parameter DOMDOM.updateSellFeeAmount(uint256,uint256)._rewardsFee (contracts/Token.sol#551) is not in mixedCase
Constant DOMDOM._totalSupply (contracts/Token.sol#258) is not in UPPER_CASE_WITH_UNDERSCORES
Constant DOMDOM.transferfee (contracts/Token.sol#259) is not in UPPER_CASE_WITH_UNDERSCORES
Constant DOMDOM.fee_denominator (contracts/Token.sol#260) is not in UPPER_CASE_WITH_UNDERSCORES
Constant DOMDOM._name (contracts/Token.sol#278) is not in UPPER_CASE_WITH_UNDERSCORES
Constant DOMDOM._symbol (contracts/Token.sol#279) is not in UPPER_CASE_WITH_UNDERSCORES
Constant DOMDOM._decimals (contracts/Token.sol#280) is not in UPPER_CASE_WITH_UNDERSCORES
Variable DOMDOM.LiquidityAdded (contracts/Token.sol#284) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#16)" inContext (contracts/Token.sol#8-19)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable IRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#112) is too similar to IRouter01.addLiquidity(address,address,
uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#113)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

DOMDOM.enableTrading() (contracts/Token.sol#568-572) uses literals with too many digits:
        - swapThreshold = (balanceOf(lpPair)) / 100000 (contracts/Token.sol#570)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

DOMDOM.LiquidityAdded (contracts/Token.sol#284) should be constant
DOMDOM.canSwapFees (contracts/Token.sol#273) should be constant
DOMDOM.maxBuyFee (contracts/Token.sol#263) should be constant
DOMDOM.maxSellFee (contracts/Token.sol#262) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

DOMDOM.swapRouter (contracts/Token.sol#277) 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 issues found

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/0x968621d7b36003d16a04e5bfffa0cfb077a8da7069db9510c0383dd6706323e5

2- Buying when excluded from fees (0% tax)(Passed):

https://testnet.bscscan.com/tx/0xe9051e37b311d522ffaa046ae5eef19a0faf20cbb300f2b2b19c7b51a9e1f6d7

3- Selling when excluded from fees (0% tax)(Passed):

https://testnet.bscscan.com/tx/0x36940f8678133e2c76512a10aa735ddc2ae926ca289fa9be3b8c15087e0fdf86

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

https://testnet.bscscan.com/tx/0x4a8191ff5b6932e079f1eecbc60e3943de2b7d84442d53964e4b0e614ef71782

5- Buying when not excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0xc13deb2256c88f6328dea4aa290a69848faa81db2c3252216d624073bddda27e

FUNCTIONAL TESTING

6- Selling when not excluded from fees (3% tax) (passed):

https://testnet.bscscan.com/tx/0xe2c9dd3aa1a12c79a7fee9691ba401d9f152ac95ac894396474241402f33cefe

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

https://testnet.bscscan.com/tx/0xd2a891522f6f647de1cca5c61b73cc392203b5c692a59011fbbbc527f7b628aa

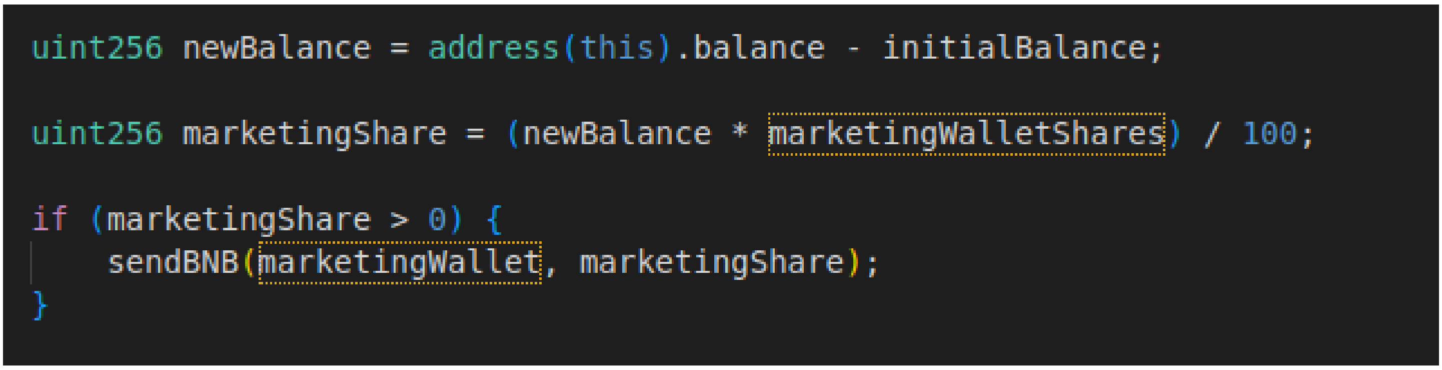
8- Internal swap (passed):

As can seen in this transaction, all the 3 fee wallets received BNB
https://testnet.bscscan.com/tx/0xe2c9dd3aa1a12c79a7fee9691ba401d9f152ac95ac894396474241402f33cefe

MANUAL TESTING

Centralization - Owner must enable trading

Severity: High
Function: enableTrading
Lines: 299-301
Status: not resolved
Overview:
The owner must activate trading for investors to buy, sell, or transfer tokens. If trading remains disabled, token holders will be unable to trade their tokens.



```
uint256 newBalance = address(this).balance - initialBalance;

uint256 marketingShare = (newBalance * marketingWalletShares) / 100;

if (marketingShare > 0) {
    sendBNB(marketingWallet, marketingShare);
}
```

Recommendation:
Incorporate a safety mechanism that allows investors to activate trading if a specified duration has elapsed since the conclusion of the presale.

MANUAL TESTING

Informative - Internal swap threshold

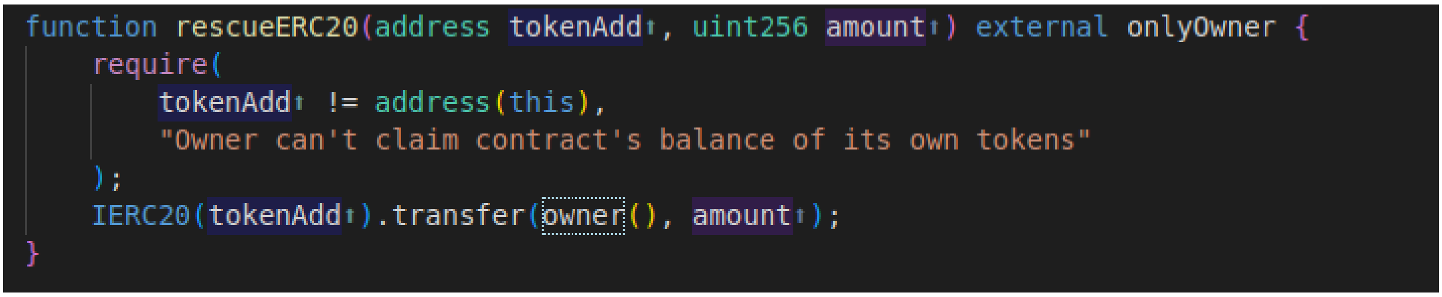
Severity: Informative
Function: --
Lines: --
Status: not resolved
Overview:
Current implemention of the contract doesn’t allow owner to change swap threshold, depending on market condition, liquidity pool size owner might need to change swap threshold accordingly.

Recommendation:
Create a function to be able to update swap threshold in a reasonable range

MANUAL TESTING

Informative - Rescuing native tokens

Severity: Informative
Function: rescueERC20
Lines: 423-426
Status: not resolved
Overview:
Currently rescueERC20 function doesn’t allow owner to withdraw native tokens. Adding this feature is suggested because removing tokens from contract doesn’t hurt trades or investors.



```
function rescueERC20(address tokenAdd, uint256 amount) external onlyOwner {
    require(
        tokenAdd != address(this),
        "Owner can't claim contract's balance of its own tokens"
    );
    IERC20(tokenAdd).transfer(owner(), amount);
}
```

Recommendation:
Allow native token withdrawals

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