AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Shibarium Classic

DATED : 14 Aug 23'

AUDIT SUMMARY

Project name – Shibarium Classic

Date: 14 Aug, 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	0	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:
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/0xbCE962f74beaacE5fCafaB6751B59603e7682Ed3

Token Information

Token Name : Shibarium Classic

Token Symbol: SHIBAC

Decimals: 18

Token Supply: 1,000,000,000

Token Address:
0xfED42E79B81cA506A79A8929C502c21E26355206

Checksum:
fda634ec90762d11c251749bc6ae5cce415f5b14

Owner:
0xE94AcfCdEcB19DF5f830D9730AF86a955F257FB7
(at time of writing the audit)

Deployer:
0xE94AcfCdEcB19DF5f830D9730AF86a955F257FB7

TOKEN OVERVIEW

Fees:

Buy Fees: 0-30%

Sell Fees: 0-30%

Transfer Fees: 0-30%

Fees Privilege: Owner

Ownership: not owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: Initial distribution of the tokens

- modifying fees

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

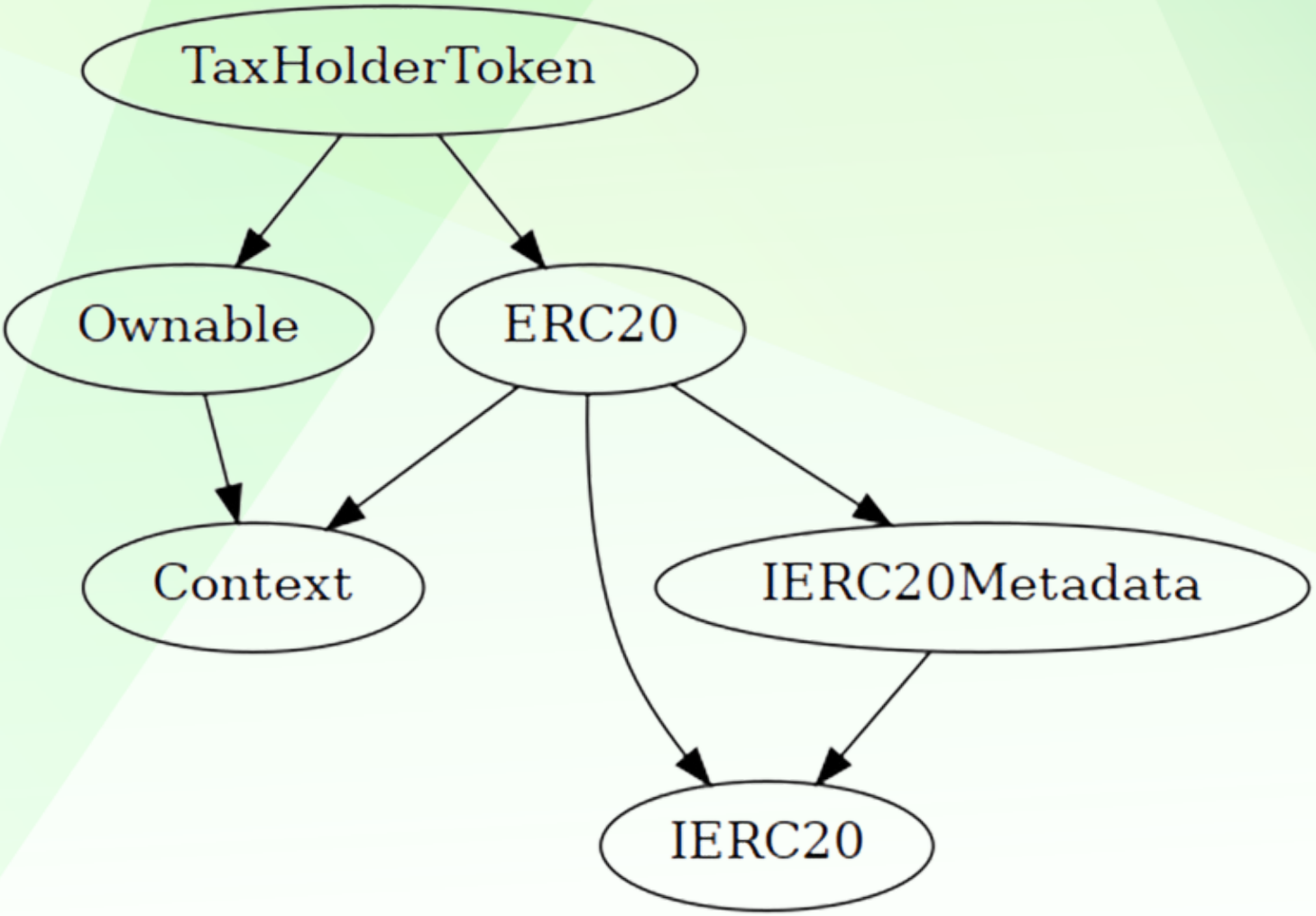
CLASSIFICATION OF RISK

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

INHERITANCE TREE



POINTS TO NOTE

- **Owner is able to update buy/sell/transfer fees (0-30%)**
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to mint new tokens
- Owner is not able to set maximum wallet and maximum buy/sell limits

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
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 ❗
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 ❗

CONTRACT ASSESMENT

└	approve	Public ❗	🔴	NO ❗
└	transferFrom	Public ❗	🔴	NO ❗
└	increaseAllowance	Public ❗	🔴	NO ❗
└	decreaseAllowance	Public ❗	🔴	NO ❗
└	_transfer	Internal 🔒	🔴	
└	_mint	Internal 🔒	🔴	
└	_burn	Internal 🔒	🔴	
└	_approve	Internal 🔒	🔴	
└	_beforeTokenTransfer	Internal 🔒	🔴	
└	_afterTokenTransfer	Internal 🔒	🔴	
TaxHolderToken	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	💵	ERC20
└	<Receive Ether>	External ❗	💵	NO ❗
└	getBalance	Private 🔐		
└	decimals	Public ❗		NO ❗
└	totalSupply	Public ❗		NO ❗
└	reflectionFee	Public ❗		NO ❗
└	getBurnFee	Public ❗		NO ❗
└	getTaxFee	Public ❗		NO ❗
└	getFeeAccount	Public ❗		NO ❗
└	isExcludedFromFee	Public ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	isExcluded	Public ❗		NO ❗
└	totalFeesRedistributed	Public ❗		NO ❗
└	excludeFromFee	Public ❗	🔴	onlyOwner
└	includeInFee	Public ❗	🔴	onlyOwner
└	changeFeeAccount	Public ❗	🔴	onlyOwner
└	changeReflectionFee	Public ❗	🔴	onlyOwner
└	changeBurnFee	Public ❗	🔴	onlyOwner
└	changeTaxFee	Public ❗	🔴	onlyOwner
└	_mintStart	Private 🔐	🔴	
└	reflect	Public ❗	🔴	NO ❗
└	reflectionFromToken	Public ❗		NO ❗
└	tokenFromReflection	Private 🔐		
└	excludeAccountFromReward	Public ❗	🔴	onlyOwner

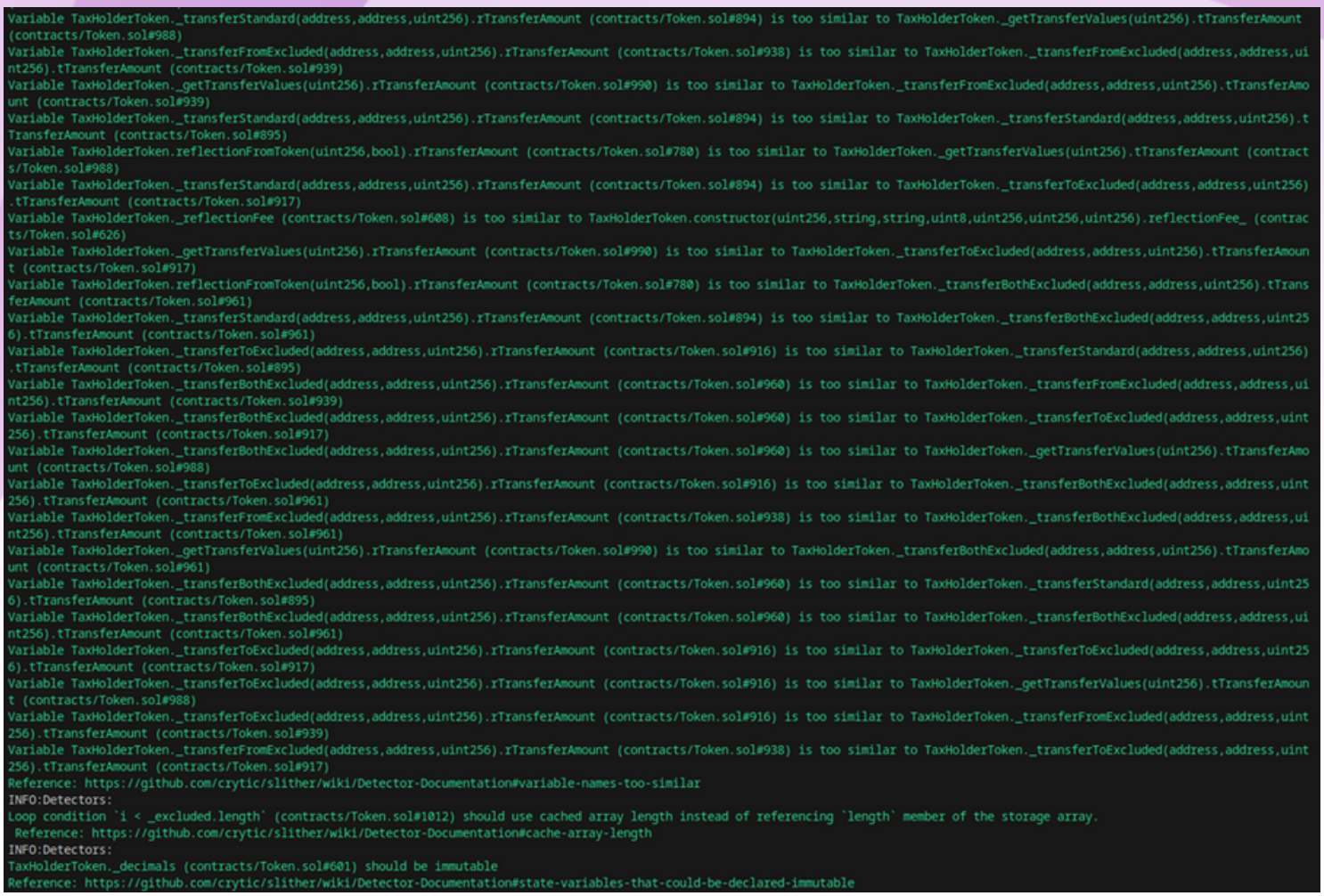
CONTRACT ASSESMENT

└	includeAccountinReward	Public ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	_getCompleteTaxValue	Private 🔐		
└	_getTransferValues	Private 🔐		
└	_reflectFee	Private 🔐	🛑	
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	burnFeeTransfer	Private 🔐	🛑	
└	taxFeeTransfer	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Variable TaxHolderToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#894) is too similar to TaxHolderToken._getTransferValues(uint256).tTransferAmount
(contracts/Token.sol#988)
Variable TaxHolderToken._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#938) is too similar to TaxHolderToken._transferFromExcluded(address,address,ui
nt256).tTransferAmount (contracts/Token.sol#939)
Variable TaxHolderToken._getTransferValues(uint256).rTransferAmount (contracts/Token.sol#990) is too similar to TaxHolderToken._transferFromExcluded(address,address,uint256).tTransferAmo
unt (contracts/Token.sol#939)
Variable TaxHolderToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#894) is too similar to TaxHolderToken._transferStandard(address,address,uint256).t
TransferAmount (contracts/Token.sol#895)
Variable TaxHolderToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#780) is too similar to TaxHolderToken._getTransferValues(uint256).tTransferAmount (contract
s/Token.sol#988)
Variable TaxHolderToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#894) is too similar to TaxHolderToken._transferToExcluded(address,address,uint256)
.tTransferAmount (contracts/Token.sol#917)
Variable TaxHolderToken._reflectionFee (contracts/Token.sol#608) is too similar to TaxHolderToken.constructor(uint256,string,string,uint8,uint256,uint256,uint256).reflectionFee_ (contrac
ts/Token.sol#626)
Variable TaxHolderToken._getTransferValues(uint256).rTransferAmount (contracts/Token.sol#990) is too similar to TaxHolderToken._transferToExcluded(address,address,uint256).tTransferAmoun
t (contracts/Token.sol#917)
Variable TaxHolderToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#780) is too similar to TaxHolderToken._transferBothExcluded(address,address,uint256).tTrans
ferAmount (contracts/Token.sol#961)
Variable TaxHolderToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#894) is too similar to TaxHolderToken._transferBothExcluded(address,address,uint25
6).tTransferAmount (contracts/Token.sol#961)
Variable TaxHolderToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#916) is too similar to TaxHolderToken._transferStandard(address,address,uint256)
.tTransferAmount (contracts/Token.sol#895)
Variable TaxHolderToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#960) is too similar to TaxHolderToken._transferFromExcluded(address,address,ui
nt256).tTransferAmount (contracts/Token.sol#939)
Variable TaxHolderToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#960) is too similar to TaxHolderToken._transferToExcluded(address,address,uint
256).tTransferAmount (contracts/Token.sol#917)
Variable TaxHolderToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#960) is too similar to TaxHolderToken._getTransferValues(uint256).tTransferAmo
unt (contracts/Token.sol#988)
Variable TaxHolderToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#916) is too similar to TaxHolderToken._transferBothExcluded(address,address,uint
256).tTransferAmount (contracts/Token.sol#961)
Variable TaxHolderToken._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#938) is too similar to TaxHolderToken._transferBothExcluded(address,address,ui
nt256).tTransferAmount (contracts/Token.sol#961)
Variable TaxHolderToken._getTransferValues(uint256).rTransferAmount (contracts/Token.sol#990) is too similar to TaxHolderToken._transferBothExcluded(address,address,uint256).tTransferAmo
unt (contracts/Token.sol#961)
Variable TaxHolderToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#960) is too similar to TaxHolderToken._transferStandard(address,address,uint25
6).tTransferAmount (contracts/Token.sol#895)
Variable TaxHolderToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#960) is too similar to TaxHolderToken._transferBothExcluded(address,address,ui
nt256).tTransferAmount (contracts/Token.sol#961)
Variable TaxHolderToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#916) is too similar to TaxHolderToken._transferToExcluded(address,address,uint25
6).tTransferAmount (contracts/Token.sol#917)
Variable TaxHolderToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#916) is too similar to TaxHolderToken._getTransferValues(uint256).tTransferAmoun
t (contracts/Token.sol#988)
Variable TaxHolderToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#916) is too similar to TaxHolderToken._transferFromExcluded(address,address,uint
256).tTransferAmount (contracts/Token.sol#939)
Variable TaxHolderToken._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#938) is too similar to TaxHolderToken._transferToExcluded(address,address,uint
256).tTransferAmount (contracts/Token.sol#917)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
Loop condition `i < _excluded.length` (contracts/Token.sol#1012) should use cached array length instead of referencing `length` member of the storage array.
 Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#cache-array-length
INFO:Detectors:
TaxHolderToken._decimals (contracts/Token.sol#601) 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

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x2247e27f193af529d2ce1034f1d210e6aa9ed7eddade812f4ad823b081436285

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x3280593c1b813931b2e1497c930278d298fef8392ef17aba6faaeef677c37eb3

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x46ae75692a8924ccd81704de6113e0f6759b6604cab642e08b1aefebce563144

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xdc191060e17660b2fdf2d53ad37e73dd7f387b808a4fd5f7c790b1680a156733

5- Buying when not excluded from fees (tax adjustable in range 0-30%) (passed):
https://testnet.bscscan.com/tx/0xcd22f2038c66a0c90d898078c6f47298d274289ef1d761893feb624936b30c4c

6- Selling when not excluded from fees (tax adjustable in range 0-30%) (passed):
https://testnet.bscscan.com/tx/0xb2e9da5d1f0a6d1cc3fcc176269e30102dedd20e29e419112ce3365aed911213

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (tax adjustable in range 0-30%) (passed):

https://testnet.bscscan.com/tx/0xa9b374ce62c360448623c4378155b485ca9b100e1729c5fdbfdda5da5d0c2ccd

Informational

Centralization – Excessive fees

Severity: Informational
function: changeBurnFee – changeTaxFee - changeReflectionFee
Status: Open
Overview:
Owner is able to set up to 30% fee on buy/sell/transfers seperatly.

```
function changeReflectionFee(
    uint256 newReflectionFee
) public onlyOwner returns (bool) {
    require(
        newReflectionFee >= 0,
        "Reflection fee must be greater or equal to zero"
    );
    require(
        newReflectionFee <= 10,
        "Reflection fee must be lower or equal to ten"
    );
    _reflectionFee = newReflectionFee;
    return true;
}

function changeBurnFee(uint256 burnFee_) public onlyOwner returns (bool) {
    require(burnFee_ >= 0, "Burn fee must be greater or equal to zero");
    require(burnFee_ <= 10, "Burn fee must be lower or equal to 10");
    _burnFee = burnFee_;
    return true;
}

function changeTaxFee(uint256 taxFee_) public onlyOwner returns (bool) {
    require(taxFee_ >= 0, "Tax fee must be greater or equal to zero");
    require(taxFee_ <= 10, "Tax fee must be lower or equal to 10");
    _taxFee = taxFee_;
    return true;
}
```

Suggestion

Maximum buy/sell/transfer fee depends on token economy and marketing condition, however as a general suggestion its a good practice to keep fees within 0-10% for buy/sell/transfers seperatly

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