AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

15thwhitepaperbtc

DATED : 01 November 23'

AUDIT SUMMARY

Project name – 15thwhitepaperbtc

Date: 01 November 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/0x5D587DbF9D1a5e A91e96e218def19B9e75810bB5

Token Information

Token Address : ---

Name: 15thwhitepaperbtc

Symbol: 15thwhitepaperbtc

Decimals: 18

Network: Binance smart chain

Token Type: BEP20

Owner: ---

Deployer: ---

Token Supply: 15

Checksum:
1666029b29a5f1ae543a23971ebc1e066fc0f1b5

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/0x5D587DbF9D1a5eA91e9 6e218def19B9e75810bB5

TOKEN OVERVIEW

buy fee: 12%

Sell fee: 12%

transfer fee: 12%

Fee Privilege: Static fees

Ownership: Owned

Minting: None

Max Tx: No

Blacklist: No

Other Privileges:

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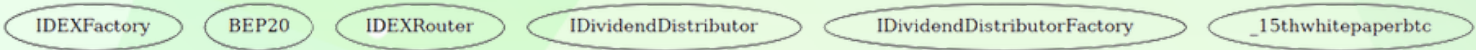
Findings

Severity	Found
◆ **Critical**	0
◆ **High-Risk**	0
◆ **Medium-Risk**	0
◆ **Low-Risk**	0
◆ **Gas Optimization / Suggestions**	1

INHERITANCE TREE

IDEXFactory BEP20 IDEXRouter IDividendDistributor IDividendDistributorFactory _15thwhitepaperbtc

POINTS TO NOTE

- Owner is not able to change current fees (12% buys and 12% sells)

- Owner is not able to blacklist an arbitrary wallet

- Owner is not able to disable trades

- Owner is not able to mint new tokens

- Owner is not able to set maxTx and max wallet

STATIC ANALYSIS

```
INFO:Detectors:
Variable _15thwhitepaperbtc.setmarketingFeeReceivers(address,address)._marketingFeeReceiver1 (contracts/Token.sol#289) is too similar to _15thwhitepaperbtc.setmarketingFe
eReceivers(address,address)._marketingFeeReceiver2 (contracts/Token.sol#290)
Variable _15thwhitepaperbtc.marketingFeeReceiver1 (contracts/Token.sol#134-135) is too similar to _15thwhitepaperbtc.marketingFeeReceiver2 (contracts/Token.sol#136-137)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
_15thwhitepaperbtc.constructor() (contracts/Token.sol#154-176) uses literals with too many digits:
        - _allowances[address(this)][address(router)] = 100000000 * (10 ** 50) * 100 (contracts/Token.sol#156-159)
_15thwhitepaperbtc.setDistributorSettings(uint256) (contracts/Token.sol#493-496) uses literals with too many digits:
        - require(bool)(gas < 3000000) (contracts/Token.sol#494)
_15thwhitepaperbtc.setTXBNBgas(uint256) (contracts/Token.sol#498-501) uses literals with too many digits:
        - require(bool)(gas < 100000) (contracts/Token.sol#499)
_15thwhitepaperbtc.setDistribuitorBuyGas(uint256) (contracts/Token.sol#503-506) uses literals with too many digits:
        - require(bool)(gas < 1000000) (contracts/Token.sol#504)
_15thwhitepaperbtc.setLiquidifyGas(uint256) (contracts/Token.sol#508-511) uses literals with too many digits:
        - require(bool)(gas < 1000000) (contracts/Token.sol#509)
_15thwhitepaperbtc.slitherConstructorVariables() (contracts/Token.sol#97-541) uses literals with too many digits:
        - distributorGas = 300000 (contracts/Token.sol#128)
_15thwhitepaperbtc.slitherConstructorVariables() (contracts/Token.sol#97-541) uses literals with too many digits:
        - distributorBuyGas = 400000 (contracts/Token.sol#130)
_15thwhitepaperbtc.slitherConstructorVariables() (contracts/Token.sol#97-541) uses literals with too many digits:
        - LiquidifyGas = 500000 (contracts/Token.sol#131)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
_15thwhitepaperbtc.DEAD (contracts/Token.sol#139) should be constant
_15thwhitepaperbtc._decimals (contracts/Token.sol#105) should be constant
_15thwhitepaperbtc._name (contracts/Token.sol#103) should be constant
_15thwhitepaperbtc._symbol (contracts/Token.sol#104) should be constant
_15thwhitepaperbtc.burnFee (contracts/Token.sol#120) should be constant
_15thwhitepaperbtc.buyTax (contracts/Token.sol#114) should be constant
_15thwhitepaperbtc.distributorFactory (contracts/Token.sol#98) should be constant
_15thwhitepaperbtc.feeDenominator (contracts/Token.sol#127) should be constant
_15thwhitepaperbtc.marketingFee (contracts/Token.sol#118) should be constant
_15thwhitepaperbtc.reflectionFee (contracts/Token.sol#119) should be constant
_15thwhitepaperbtc.router (contracts/Token.sol#100) should be constant
_15thwhitepaperbtc.sellBurnFee (contracts/Token.sol#125) should be constant
_15thwhitepaperbtc.sellMarketingFee (contracts/Token.sol#123) should be constant
_15thwhitepaperbtc.sellReflectionFee (contracts/Token.sol#124) should be constant
_15thwhitepaperbtc.sellTax (contracts/Token.sol#115) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
_15thwhitepaperbtc.WBNB (contracts/Token.sol#101) should be immutable
_15thwhitepaperbtc.distributor (contracts/Token.sol#99) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Token.sol analyzed (6 contracts with 88 detectors), 65 result(s) found
```

Result => A static analysis of contract's source code has been performed using slither,
No major issues were found in the output

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IDEXFactory	Interface			
└	createPair	External ❗	🔴	NO ❗
BEP20	Interface			
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🔴	NO ❗
└	approve	External ❗	🔴	NO ❗
└	transferFrom	External ❗	🔴	NO ❗
IDEXRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	getAmountsOut	External ❗		NO ❗
IDividendDistributor	Interface			
└	setDistributionCriteria	External ❗	🔴	NO ❗
└	setShare	External ❗	🔴	NO ❗
└	deposit	External ❗	💵	NO ❗
└	process	External ❗	🔴	NO ❗
└	claimDividend	External ❗	🔴	NO ❗
└	getUnpaidEarnings	External ❗		NO ❗
IDividendDistributorFactory	Interface			
└	createDistribuitor	External ❗	🔴	NO ❗

CONTRACT ASSESMENT

_15thwhitepaperbtc	Implementation			
└	<Constructor>	Public ❗	💵	NO❗
└	<Receive Ether>	External ❗	💵	NO❗
└	totalSupply	External ❗		NO❗
└	owner	Public ❗		NO❗
└	decimals	External ❗		NO❗
└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	allowance	External ❗		NO❗
└	transfer	External ❗	🔴	NO❗
└	approve	Public ❗	🔴	NO❗
└	transferFrom	External ❗	🔴	NO❗
└	setPair	Public ❗	🔴	onlyOwner
└	excludeFromFee	Public ❗	🔴	onlyOwner
└	includeInFee	Public ❗	🔴	onlyOwner
└	setDividendExempt	Public ❗	🔴	onlyOwner
└	isExcludedFromFee	Public ❗		NO❗
└	burn	External ❗	🔴	NO❗
└	_burn	Internal 🔒	🔴	
└	_burnIN	Internal 🔒	🔴	
└	shouldSwapBack	Internal 🔒		
└	setmarketingFeeReceivers	External ❗	🔴	onlyOwner
└	setSwapBackSettings	External ❗	🔴	onlyOwner
└	value	Public ❗		NO❗
└	_isSell	Internal 🔒		
└	fee	Internal 🔒		
└	_transferFrom	Internal 🔒	🔴	
└	_basicTransfer	Internal 🔒	🔴	
└	_txTransfer	Internal 🔒	🔴	
└	getamount	Internal 🔒		
└	swapBack	Internal 🔒	🔴	swapping
└	manualSend	External ❗	🔴	onlyOwner
└	setDistributionCriteria	External ❗	🔴	onlyOwner
└	claimDividend	External ❗	🔴	NO❗
└	getUnpaidEarnings	Public ❗		NO❗
└	setDistributorSettings	External ❗	🔴	onlyOwner

CONTRACT ASSESMENT

└	setTXBNBgas	External ❗	🔴	onlyOwner
└	setDistribuitorBuyGas	External ❗	🔴	onlyOwner
└	setLiquidifyGas	External ❗	🔴	onlyOwner
└	renounceOwnership	Public ❗	🔴	onlyOwner
└	transferOwnership	Public ❗	🔴	onlyOwner
└	_transferOwnership	Internal 🔒	🔴	

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

FUNCTIONAL TESTING

1- Adding liquidity (passed**):**

https://testnet.bscscan.com/tx/0xe30147afe03b811f6dff95b4eaec340ea783bfe2df2
3524c4919ce24f4e9f902

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

https://testnet.bscscan.com/tx/0x5c86e5cb408855cccb76deabb19d0e232c607dd40
716fc0ad5022ab64de5a32c

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

https://testnet.bscscan.com/tx/0x1ac518d8edab7ec0140682c020705b54d6e6823f6
d4c9dce9d39c41f3d3912fd

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

https://testnet.bscscan.com/tx/0x9d15d1e7e44d6298773455511e053e2722f083e2d2
aa918280200870b0e1fa6d3

5- Buying when not excluded from fees (tax 12%) (passed**):**

https://testnet.bscscan.com/tx/0x5cb7da3d73b7539d1cedddf62f3d39261b03d6385
44fb7b460cc0df98e5cab25

6- Selling when not excluded from fees (tax 12%) (passed**):**

https://testnet.bscscan.com/tx/0x8d6d814ad7551d23bba0de50185bbb74940ffa458
1000477a8c497ca4d206164

7- Transferring when not excluded from fees (12 % tax) (passed**):**

https://testnet.bscscan.com/tx/0x45df08cc55ba079a709ec94fa2fb44716bf0470c0a
5900c174018b904580e97f

7- Internal swap (Marketing fee + reflections) (passed**):**

https://testnet.bscscan.com/tx/0xbb76abc5964110ba447863e4a4d8456bb52dbdd2
eaf368702985cb0b1a2e4a82

MANUAL TESTING

Logical – Out of scope dividend tracker
Severity: Unknown
function: ---
Status: Open

Overview:
Contract is using an out of scope and unverified dividend tracker. Dividend tracker is responsible for keeping track of each user rewards share and distributing BTC rewards, any mis-functioning or malicious activity related to dividend tracker which may affect buy/sell/transfer transaction is out of scope of this audit report

Suggestion
Provide the dividend tracker contract for security analysis

DISCLAIMER

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