auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Your Wallet

DATED : 25 JAN 23'

AUDIT SUMMARY

Project name – Your Wallet

Date: 25 January , 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: Passed (Contract is developed by Pinksale safu dev

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	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 done on BSC Test 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/token/0x35cbeeB6c3D960A35637E810e8676B4668c5912A#readContract

Token Information

Token Name : YourWallet

Token Symbol: YourWallet

Decimals: 18

Token Supply: 100,000,000

Token Address:
0x04Cc8c50a7dC741A2FdCb63DC58FD8b128E3EbBb

Checksum:
CE9559EA44723E2B93C915DE8487EB6A888D4F86147BB04EF6B0F165DDFF9FCD

Owner:
0x2e76Af6DDD9C3314BC7f8ed6BB4690326731301e

TOKEN OVERVIEW

Fees:

Buy Fees: 2%

Sell Fees: 2%

Transfer Fees: 0%

Fees Privilige: None

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: No

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

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

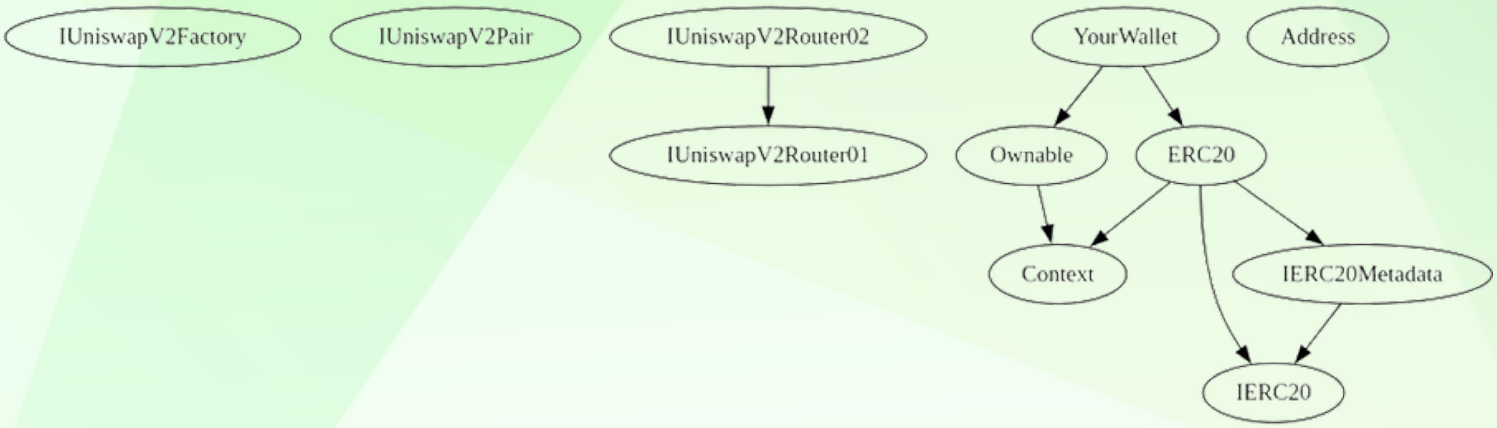
CLASSIFICATION OF RISK

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	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to change taxes (2% buy and 2% sell static, 0% transfer tax)
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to set max buy/sell/transfer amounts
- 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**
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

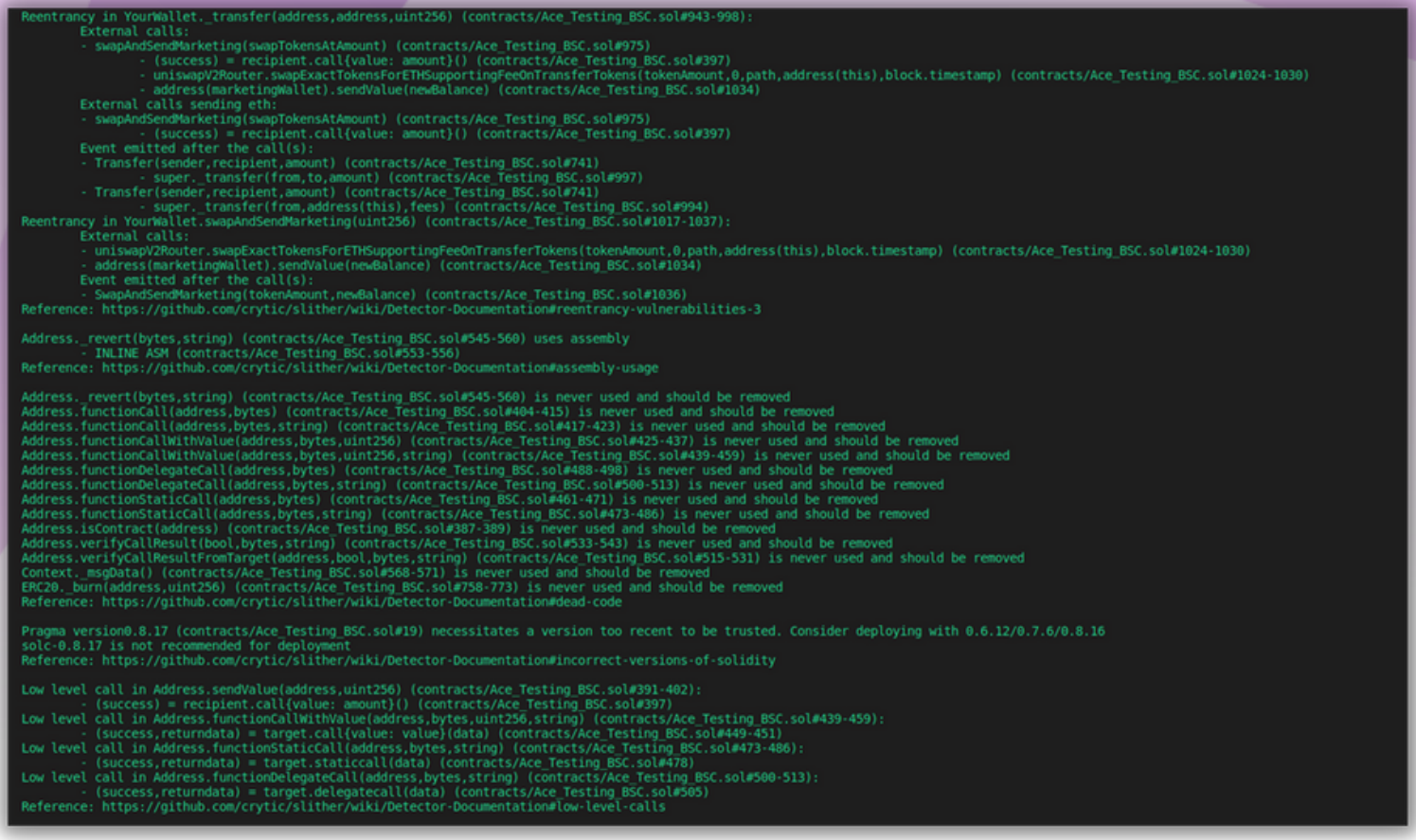
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	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
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 🔒	🛑	

CONTRACT ASSESMENT

└	_approve	Internal 🔒	🛑		
└	_beforeTokenTransfer	Internal 🔒	🛑		
└	_afterTokenTransfer	Internal 🔒	🛑		
YourWallet	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🛑	ERC20
└	<Receive Ether>	External ❗	💵	NO❗
└	claimStuckTokens	External ❗	🛑	onlyOwner
└	excludeFromFees	External ❗	🛑	onlyOwner
└	isExcludedFromFees	Public ❗		NO❗
└	changeMarketingWallet	External ❗	🛑	onlyOwner
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑		
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	swapAndSendMarketing	Private 🔐	🛑		

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



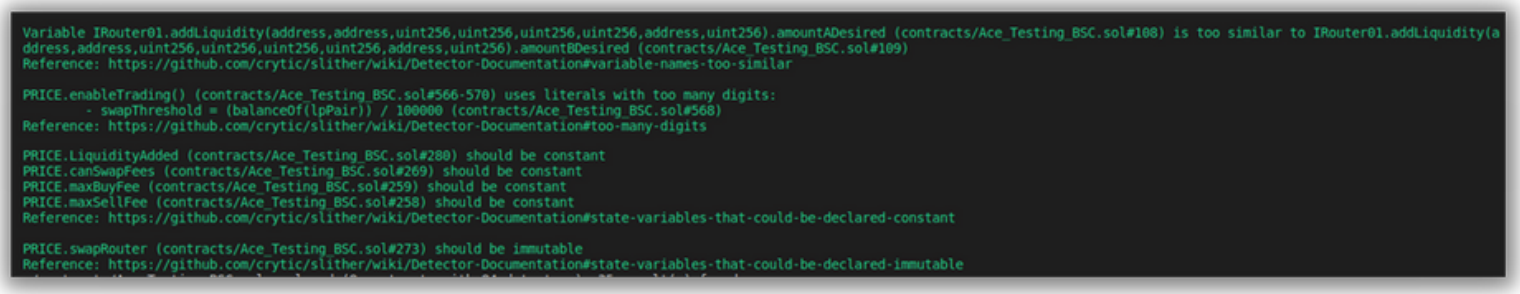
```
Reentrancy in YourWallet._transfer(address,address,uint256) (contracts/Ace_Testing_BSC.sol#943-998):
        External calls:
        - swapAndSendMarketing(swapTokensAtAmount) (contracts/Ace_Testing_BSC.sol#975)
                - (success) = recipient.call{value: amount}() (contracts/Ace_Testing_BSC.sol#397)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Ace_Testing_BSC.sol#1024-1030)
                - address(marketingWallet).sendValue(newBalance) (contracts/Ace_Testing_BSC.sol#1034)
        External calls sending eth:
        - swapAndSendMarketing(swapTokensAtAmount) (contracts/Ace_Testing_BSC.sol#975)
                - (success) = recipient.call{value: amount}() (contracts/Ace_Testing_BSC.sol#397)
        Event emitted after the call(s):
        - Transfer(sender,recipient,amount) (contracts/Ace_Testing_BSC.sol#741)
                - super._transfer(from,to,amount) (contracts/Ace_Testing_BSC.sol#997)
        - Transfer(sender,recipient,amount) (contracts/Ace_Testing_BSC.sol#741)
                - super._transfer(from,address(this),fees) (contracts/Ace_Testing_BSC.sol#994)
Reentrancy in YourWallet.swapAndSendMarketing(uint256) (contracts/Ace_Testing_BSC.sol#1017-1037):
        External calls:
        - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Ace_Testing_BSC.sol#1024-1030)
        - address(marketingWallet).sendValue(newBalance) (contracts/Ace_Testing_BSC.sol#1034)
        Event emitted after the call(s):
        - SwapAndSendMarketing(tokenAmount,newBalance) (contracts/Ace_Testing_BSC.sol#1036)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address._revert(bytes,string) (contracts/Ace_Testing_BSC.sol#545-560) uses assembly
        - INLINE ASM (contracts/Ace_Testing_BSC.sol#553-556)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address._revert(bytes,string) (contracts/Ace_Testing_BSC.sol#545-560) is never used and should be removed
Address.functionCall(address,bytes) (contracts/Ace_Testing_BSC.sol#404-415) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/Ace_Testing_BSC.sol#417-423) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/Ace_Testing_BSC.sol#425-437) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Ace_Testing_BSC.sol#439-459) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/Ace_Testing_BSC.sol#488-498) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/Ace_Testing_BSC.sol#500-513) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/Ace_Testing_BSC.sol#461-471) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/Ace_Testing_BSC.sol#473-486) is never used and should be removed
Address.isContract(address) (contracts/Ace_Testing_BSC.sol#387-389) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/Ace_Testing_BSC.sol#533-543) is never used and should be removed
Address.verifyCallResultFromTarget(address,bool,bytes,string) (contracts/Ace_Testing_BSC.sol#515-531) is never used and should be removed
Context._msgData() (contracts/Ace_Testing_BSC.sol#568-571) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Ace_Testing_BSC.sol#758-773) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.17 (contracts/Ace_Testing_BSC.sol#19) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.17 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/Ace_Testing_BSC.sol#391-402):
        - (success) = recipient.call{value: amount}() (contracts/Ace_Testing_BSC.sol#397)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Ace_Testing_BSC.sol#439-459):
        - (success,returndata) = target.call{value: value}(data) (contracts/Ace_Testing_BSC.sol#449-451)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Ace_Testing_BSC.sol#473-486):
        - (success,returndata) = target.staticcall(data) (contracts/Ace_Testing_BSC.sol#478)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Ace_Testing_BSC.sol#500-513):
        - (success,returndata) = target.delegatecall(data) (contracts/Ace_Testing_BSC.sol#505)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```



```
Variable IRouter01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Ace_Testing_BSC.sol#108) is too similar to IRouter01.addLiquidity(a
ddress,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Ace_Testing_BSC.sol#109)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

PRICE.enableTrading() (contracts/Ace_Testing_BSC.sol#566-570) uses literals with too many digits:
        - swapThreshold = (balanceOf(lpPair)) / 100000 (contracts/Ace_Testing_BSC.sol#568)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

PRICE.LiquidityAdded (contracts/Ace_Testing_BSC.sol#280) should be constant
PRICE.canSwapFees (contracts/Ace_Testing_BSC.sol#269) should be constant
PRICE.maxBuyFee (contracts/Ace_Testing_BSC.sol#259) should be constant
PRICE.maxSellFee (contracts/Ace_Testing_BSC.sol#258) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

PRICE.swapRouter (contracts/Ace_Testing_BSC.sol#273) 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 (except some "minor impact" suggestions)

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

0- Deploying (Passed):

https://testnet.bscscan.com/tx/0x6deaadec8164053f3c3280242e181340ee20f6873b7584851e180ebb04761e9f

1- Adding Liquidity (Passed):
liquidity added on Pancakeswap V2:

https://testnet.bscscan.com/tx/0x384a66401ed6a2f7aecab97a6e2afaf6cf25284058fe2429a8cb911f4d70ba8c

no issue were found on adding liquidity.

2- Enabling trade for public (Passed):

https://testnet.bscscan.com/tx/0x209b9f0f6a162f6c7643485a2ff2747f31f4b0eb0a4a7f5380e794558e3a8c17

3- Excluding deployer's wallet from taxes (to test taxes) (Passed):

https://testnet.bscscan.com/tx/0x4259b5c1978b258746e3582c0dfae7de2b15220111b9a150f9d0a5b33707a4b6

FUNCTIONAL TESTING

4- Buying from a non-excluded wallet (Passed):

https://testnet.bscscan.com/tx/0xa4ba3ac8ad67dd409775614ec1485210e7f2245bb3a96addec5f946c116a0cbf

2% static tax on buys (sent to contract)

5- Selling from a non-excluded wallet (Passed):

https://testnet.bscscan.com/tx/0x5f0eff4211d81b2f514ea04b421a0b69cfbe3c9d17238a4f42fe07cf132185a9

2% static tax on sells

6- Swap & liquifiy (Passed):

taxes were swapped to BNB and sent to marketing wallets.

https://testnet.bscscan.com/address/0xbf21ce7e474e7d447cdced111de8f3ae87822ed6#internaltx

MANUAL TESTING

NO ISSUES WERE FOUND

Social Media Overview

Here are the Social Media Accounts of

Your Wallet

https://t.me/YourWallet_Official

https://twitter.com/YourWalletLive

https://yourwallet.live/

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