auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

PIKACHU

DATED : 16 MAY 23'

AUDIT SUMMARY

Project name – PIKACHU

Date: 16 May, 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	2
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1.Manual Review: The code has undergone a line-by-line review by the **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/0xD1c34E46aD55DA2a7bb86EA32a48C15d8d1Ef9CF

Payments Made :

https://bscscan.com/tx/0xbef93ec2e2ceb8742da080153ee186d077d9a0fb808bac3ccdd2a5da487f0c2a

Token Information

Name : Pikachu Inu

Symbol : PIKACHU

Decimals: 9

Network: BSC

Token Type: BEP20

Token Address:
0x21071e9F38C01FA3Ad46D4f77812c99e5c722CAD

Owner:
0xE732Db3E12953b5d7b0C077d429927b5B3c85682 **(at time of writing the audit)**

Deployer:0xE732Db3E12953b5d7b0C077d429927b5B3c85682

Token Information

Fees:

Buy Fees: 10%

Sell Fees: 10%

Transfer Fees: 10%

Fees Privilige: None

Ownership : Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: Including and excluding form 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

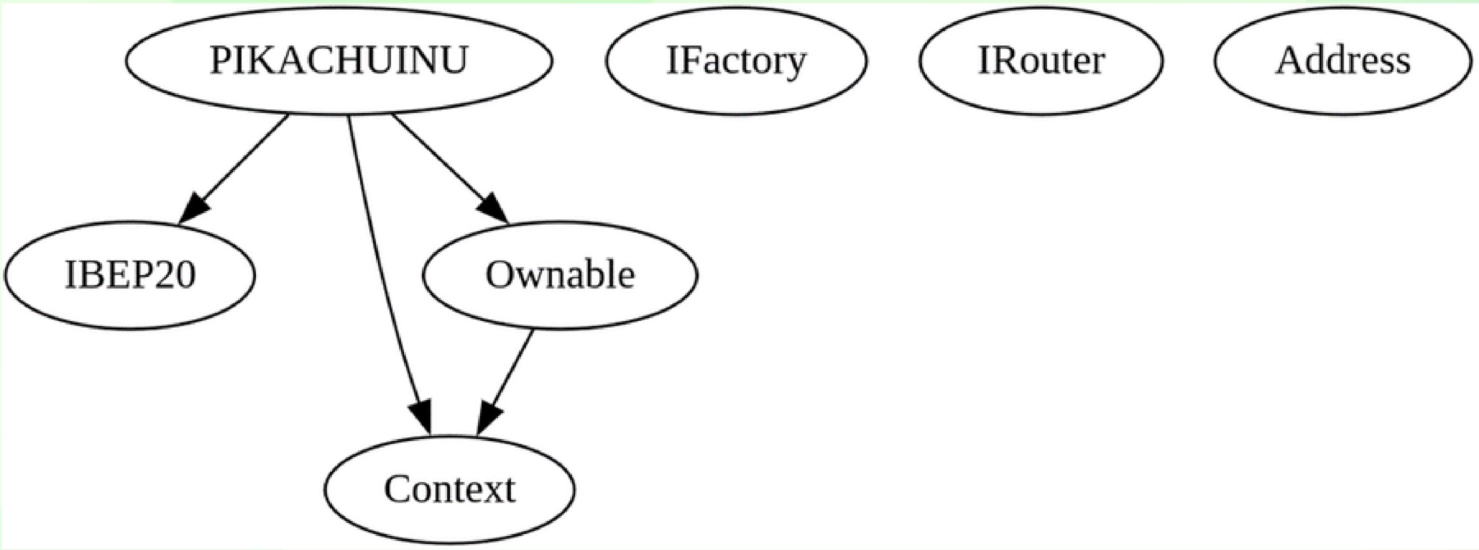
CLASSIFICATION OF RISK

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	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	2

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to modify fees (10% buy/sell/transfers)
- Owner must enable trading for investors to be able to trade
- 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**  |
||||||
| **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 |
| └ | _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 🔒 | 🛑  | |
||||||
| **PIKACHUINU** | 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❗️ |
```

CONTRACT ASSESMENT

```
| └ | approve | Public ❗ | 🛑 |NO❗ |
| └ | transferFrom | Public ❗ | 🛑 |NO❗ |
| └ | increaseAllowance | Public ❗ | 🛑 |NO❗ |
| └ | decreaseAllowance | Public ❗ | 🛑 |NO❗ |
| └ | transfer | Public ❗ | 🛑 |NO❗ |
| └ | isExcludedFromReward | Public ❗ | |NO❗ |
| └ | reflectionFromToken | Public ❗ | |NO❗ |
| └ | tokenFromReflection | Public ❗ | |NO❗ |
| └ | excludeFromReward | Public ❗ | 🛑 | onlyOwner |
| └ | includeInReward | External ❗ | 🛑 | onlyOwner |
| └ | excludeFromFee | Public ❗ | 🛑 | onlyOwner |
| └ | includeInFee | Public ❗ | 🛑 | onlyOwner |
| └ | isExcludedFromFee | Public ❗ | |NO❗ |
| └ | _reflectRfi | Private 🔐 | 🛑 | |
| └ | _takeMarketing | Private 🔐 | 🛑 | |
| └ | _getValues | Private 🔐 | | |
| └ | _getTValues | Private 🔐 | | |
| └ | _getRValues | Private 🔐 | | |
| └ | _getRate | Private 🔐 | | |
| └ | _getCurrentSupply | Private 🔐 | | |
| └ | _approve | Private 🔐 | 🛑 | |
| └ | _transfer | Private 🔐 | 🛑 | |
| └ | _tokenTransfer | Private 🔐 | 🛑 | |
| └ | swapAndLiquify | Private 🔐 | 🛑 | lockTheSwap |
| └ | swapTokensForBNB | Private 🔐 | 🛑 | |
| └ | bulkExcludeFee | External ❗ | 🛑 | onlyOwner |
| └ | <Receive Ether> | External ❗ | 💵 |NO❗ |
||||||
| **DividendPayingToken** | Implementation | ERC20, DividendPayingTokenInterface, Ownable |||
| └ | <Constructor> | Public ❗ | 🛑 | ERC20 |
| └ | <Receive Ether> | External ❗ | 💵 |NO❗ |
| └ | distributeDividends | Public ❗ | 💵 |NO❗ |
| └ | _withdrawDividendOfUser | Internal 🔒 | 🛑 | |
| └ | setRewardToken | External ❗ | 🛑 | onlyOwner |
| └ | swapBnbForCustomToken | Internal 🔒 | 🛑 | |
| └ | dividendOf | Public ❗ | |NO❗ |
| └ | withdrawableDividendOf | Public ❗ | |NO❗ |
| └ | withdrawnDividendOf | Public ❗ | |NO❗ |
| └ | accumulativeDividendOf | Public ❗ | |NO❗ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | _tokengeneration | Internal 🔒 | 🛑 | |
```

CONTRACT ASSESMENT

```
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _setBalance | 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 ❗️ |
| └ | approve | Public ❗️ | 🛑  |NO ❗️ |
| └ | transferFrom | Public ❗️ | 🛑  |NO ❗️ |
| └ | increaseAllowance | Public ❗️ | 🛑  |NO ❗️ |
| └ | decreaseAllowance | Public ❗️ | 🛑  |NO ❗️ |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _tokengeneration | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |
||||||
| **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 🔒 |   | |
||||||
| **SafeMath** | Library |  |||
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
```

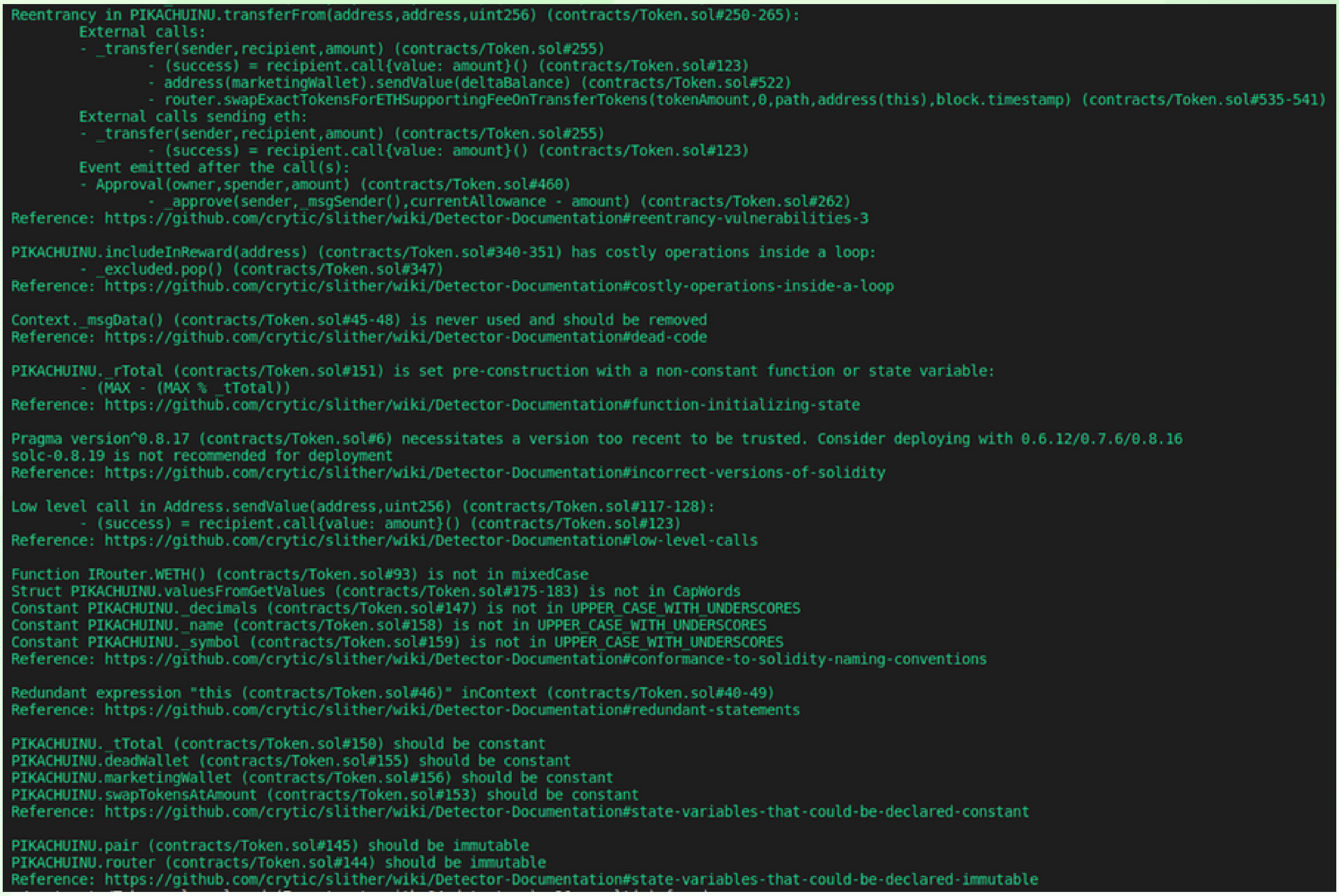
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
SafeMathInt	Library			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		
└	toUint256Safe	Internal 🔒		
SafeMathUint	Library			
└	toInt256Safe	Internal 🔒		
DividendPayingTokenInterface	Interface			
└	dividendOf	External ❗		NO ❗
└	distributeDividends	External ❗	💵	NO ❗
└	withdrawableDividendOf	External ❗		NO ❗
└	withdrawnDividendOf	External ❗		NO ❗
└	accumulativeDividendOf	External ❗		NO ❗
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IPair	Interface			
└	sync	External ❗	🛑	NO ❗
IFactory	Interface			
└	createPair	External ❗	🛑	NO ❗
└	getPair	External ❗		NO ❗
IRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗

└	swapExactETHForTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Reentrancy in PIKACHUINU.transferFrom(address,address,uint256) (contracts/Token.sol#250-265):
        External calls:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#255)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
                - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#522)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#535-541)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#255)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (contracts/Token.sol#460)
                - _approve(sender,_msgSender(),currentAllowance - amount) (contracts/Token.sol#262)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

PIKACHUINU.includeInReward(address) (contracts/Token.sol#340-351) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#347)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#45-48) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

PIKACHUINU._rTotal (contracts/Token.sol#151) 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#6) 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 Address.sendValue(address,uint256) (contracts/Token.sol#117-128):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#93) is not in mixedCase
Struct PIKACHUINU.valuesFromGetValues (contracts/Token.sol#175-183) is not in CapWords
Constant PIKACHUINU._decimals (contracts/Token.sol#147) is not in UPPER_CASE_WITH_UNDERSCORES
Constant PIKACHUINU._name (contracts/Token.sol#158) is not in UPPER_CASE_WITH_UNDERSCORES
Constant PIKACHUINU._symbol (contracts/Token.sol#159) 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#46)" inContext (contracts/Token.sol#40-49)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

PIKACHUINU._tTotal (contracts/Token.sol#150) should be constant
PIKACHUINU.deadWallet (contracts/Token.sol#155) should be constant
PIKACHUINU.marketingWallet (contracts/Token.sol#156) should be constant
PIKACHUINU.swapTokensAtAmount (contracts/Token.sol#153) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

PIKACHUINU.pair (contracts/Token.sol#145) should be immutable
PIKACHUINU.router (contracts/Token.sol#144) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Static Analysis

an static analysis of the code were performed using slither. No issues were 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/0x0441a076db0d74dad7dd62d141016f80b5650cbbad1599f0f49bee179fdb53f2

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x7eac4b26887170295ab7b7b562e65b582e00c73909a457019e44c1768d5b0f01

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x76b5ca9d605ff5454b8943dedcad3bfb01c1c4752d83bc1dcbf5c765b432d589

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x2e28f782399d6f863d0db17cfbe80aca505215bf7e5e6870a3c9e8845c15cc2c

5- Buying when not excluded from fees (10% tax) (passed):
https://testnet.bscscan.com/tx/0x5cb0f9e10e398faedf7a51c5d2b683714dd6982daa3af13a25bd9aaa3cc90c6d

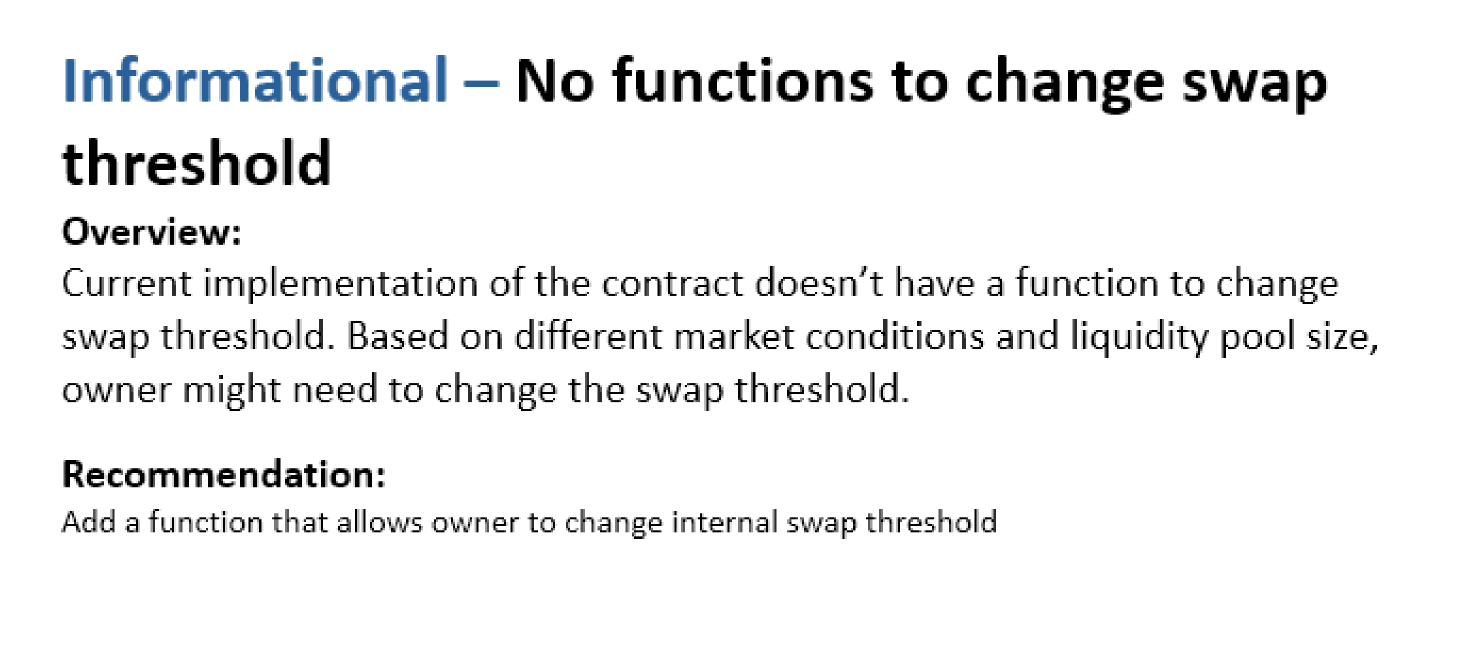
6- Selling when not excluded from fees (10% tax) (passed):
https://testnet.bscscan.com/tx/0xc0c6b49c2525ceae65ba599cc794e95d411896d1d1c16d515fe8138f04ea5ad8

7- Transferring when not excluded from fees (10% tax) (passed):
https://testnet.bscscan.com/tx/0x8fc4f551c966574d8f9071b473e7afb490d02e72e690bd40d0e8e109567b99dc

FUNCTIONAL TESTING

8- Internal swap (passed):
marketing wallet received Rewards
https://testnet.bscscan.com/token/0xD1c34E46aD55DA2a7bb86EA32a48C15d8d1Ef9CF

FUNCTIONAL TESTING



Informational – No functions to change swap threshold

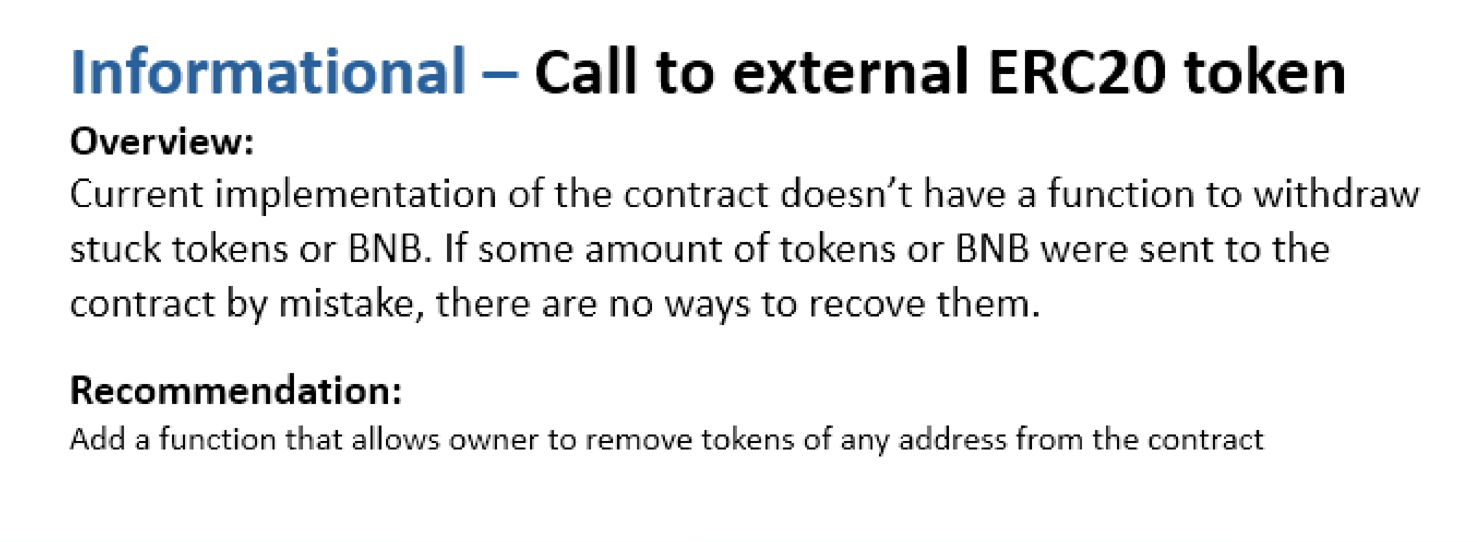
Overview:

Current implementation of the contract doesn’t have a function to change swap threshold. Based on different market conditions and liquidity pool size, owner might need to change the swap threshold.

Recommendation:

Add a function that allows owner to change internal swap threshold

FUNCTIONAL TESTING



Informational – Call to external ERC20 token

Overview:

Current implementation of the contract doesn’t have a function to withdraw stuck tokens or BNB. If some amount of tokens or BNB were sent to the contract by mistake, there are no ways to recove them.

Recommendation:

Add a function that allows owner to remove tokens of any address from the contract

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