auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

POPE

DATED : 19 June 23'

AUDIT SUMMARY

Project name – POPE

Date: 19 June, 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	3
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/0xd70294601618db7bc80d35d6145a5ad9c9185e27

Token Information

Token Name : POPEYE

Token Symbol: POPE

Decimals: 9

Token Supply: 100,000,000,000

Token Address:
0xA2D46980472e17440f8448A00E401846a3F6cC31

Checksum:
e64d7e9d0a671844d201c5a2f59544adfd14c2df

Owner:
0xf5F2a0255310F97eda5ed25D6cB23c34e6a1B5Ea
(at time of writing the audit)

Deployer:
0xf5F2a0255310F97eda5ed25D6cB23c34e6a1B5Ea

TOKEN OVERVIEW

Fees:

Buy Fees: 5%

Sell Fees: 5%

Transfer Fees: 5%

Fees Privilege: Static fees

Ownership: owned

Minting: none

Max Tx Amount/ Max Wallet Amount: 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
- ✓ 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

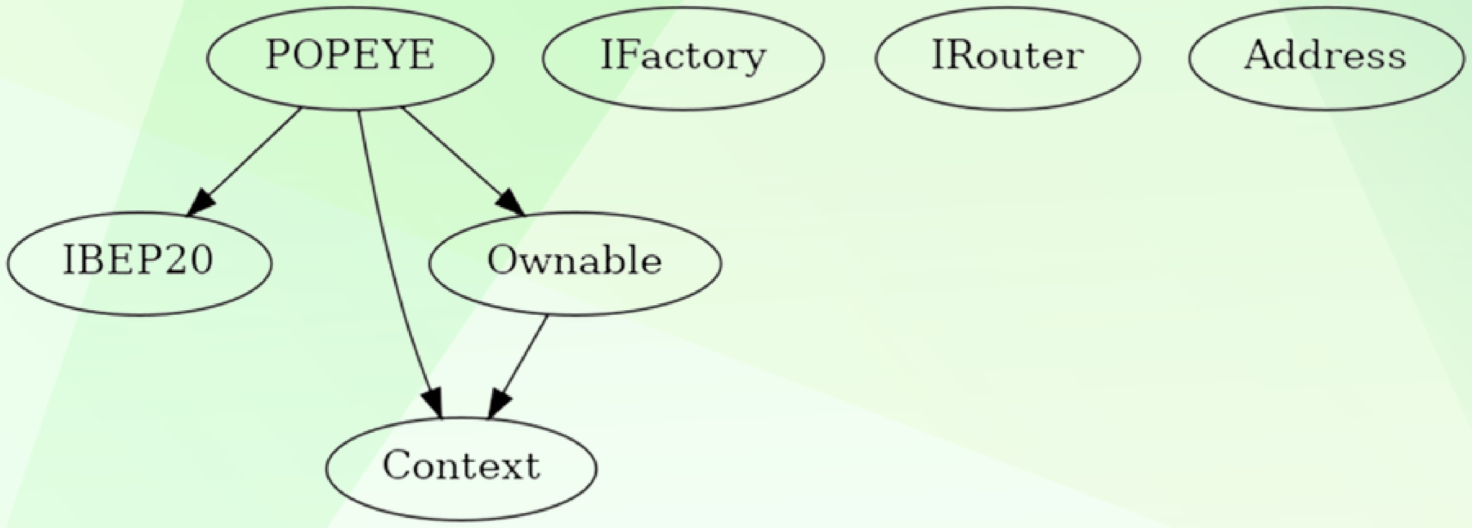
CLASSIFICATION OF RISK

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	3

INHERITANCE TREE



POINTS TO NOTE

- owner is not able to change fees (5% tax on transfer/buy/sell)
- owner is not able to set max buy/sell/transfer/wallet limits
- owner is not able to blacklist an arbitrary wallet
- owner is not able to mint new tokens
- owner is not able to disable trades

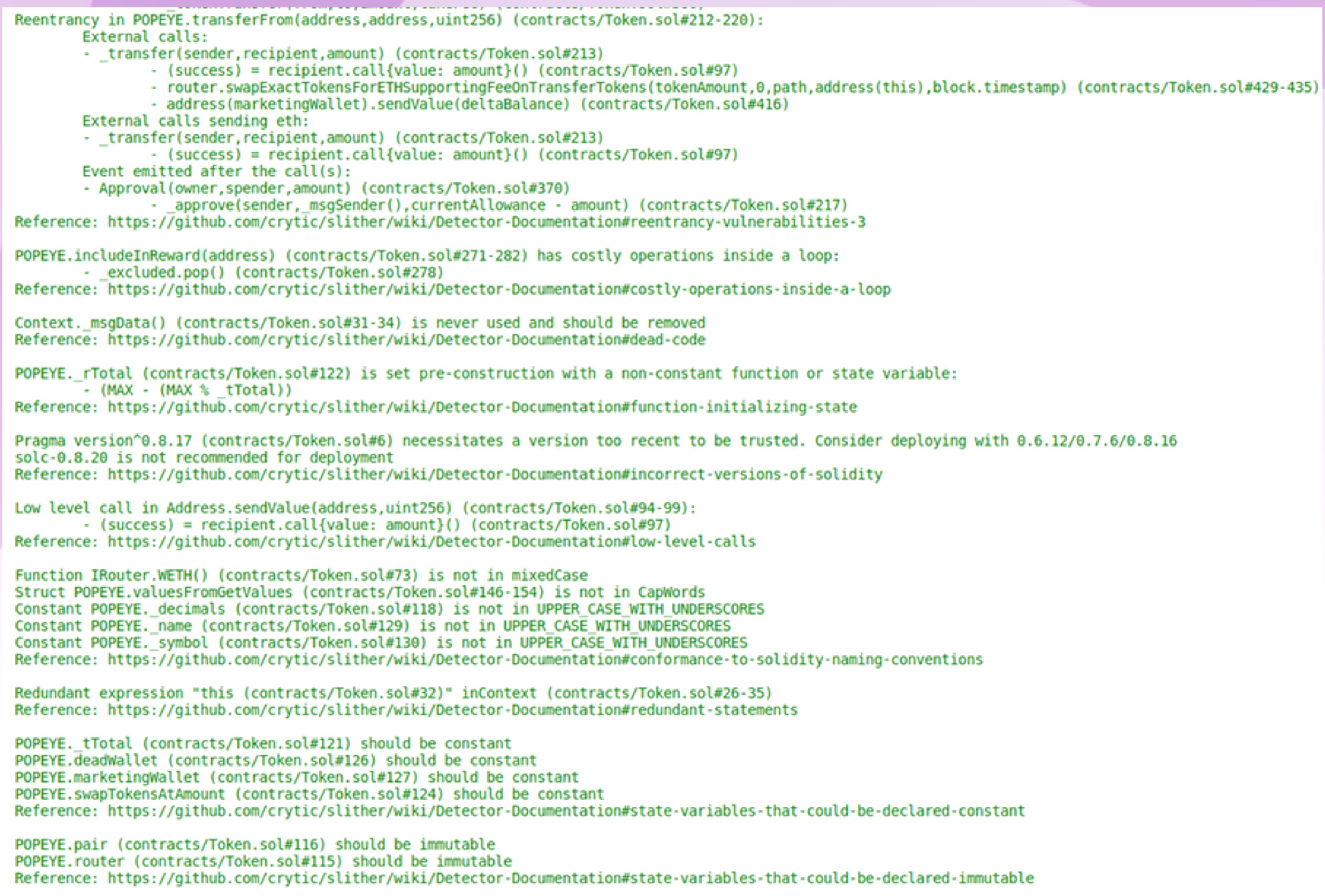
CONTRACT ASSESMENT

```
| 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 🔒 | 🛑  | |
||||||
| **POPEYE** | 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❗️ |
| └ | 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 ❗ |


### Legend

| Symbol | Meaning |
|:--------:|-----------|
| 🛑 | Function can modify state |
| 💵 | Function is payable |
```

STATIC ANALYSIS



```
Reentrancy in POPEYE.transferFrom(address,address,uint256) (contracts/Token.sol#212-220):
        External calls:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#213)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#97)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#429-435)
                - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#416)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#213)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#97)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (contracts/Token.sol#370)
                - _approve(sender,_msgSender(),currentAllowance - amount) (contracts/Token.sol#217)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

POPEYE.includeInReward(address) (contracts/Token.sol#271-282) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#278)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#31-34) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

POPEYE._rTotal (contracts/Token.sol#122) 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-0.8.20 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#94-99):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#97)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#73) is not in mixedCase
Struct POPEYE.valuesFromGetValues (contracts/Token.sol#146-154) is not in CapWords
Constant POPEYE._decimals (contracts/Token.sol#118) is not in UPPER_CASE_WITH_UNDERSCORES
Constant POPEYE._name (contracts/Token.sol#129) is not in UPPER_CASE_WITH_UNDERSCORES
Constant POPEYE._symbol (contracts/Token.sol#130) 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#32)" inContext (contracts/Token.sol#26-35)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

POPEYE._tTotal (contracts/Token.sol#121) should be constant
POPEYE.deadWallet (contracts/Token.sol#126) should be constant
POPEYE.marketingWallet (contracts/Token.sol#127) should be constant
POPEYE.swapTokensAtAmount (contracts/Token.sol#124) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

POPEYE.pair (contracts/Token.sol#116) should be immutable
POPEYE.router (contracts/Token.sol#115) 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/0x15f5868c64308b62fb56e35a09c2a2c17ebc8e3dd12282e1aa10daefdc3505c2

2- Buying when excluded from fees (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0xac7eaedf458de1f9499b38e5775c40f634d81cc578a5db8c892fa2ffbdba6aa7

3- Selling when excluded from fees (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0xff33c46e1ad798f6e97bfb248ed8ba9c291d088cf9c5bedf1d2ad69532fe6778

4- Transferring when excluded from fees (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0xc0a302bf7940d5f7fd553c08025d5d196e47fd5c2ffffbb0f0d5dd72e12600d0

5- Buying when not excluded from fees (5% tax) **(passed):**
https://testnet.bscscan.com/tx/0x21613d09fef93f1883dbe6a9e71da39008806b6908a6fbcfa3d9463d5f56015c

6- Selling when not excluded from fees (5% tax) **(passed):**
https://testnet.bscscan.com/tx/0x496e43ec2031d90ee0ec0c5bcd4c30d3388b3bd44ffc189d6c7acbf518042bda

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (5% tax) **(passed):**
https://testnet.bscscan.com/tx/0xdfadaf6e7bd656e991c26ba9007a4131c3e56a3e5431d7f4a02379c5ac334270

8- Internal swap **(passed):**
- BNB fee sent to marketing wallet
https://testnet.bscscan.com/address/0xd224be5904ac735928b5955e80620f9a32090b6f#internaltx

FUNCTIONAL TESTING

Missing Logic – Stuck ETH and Tokens

Severity: Informational

Status: Open

Overview:

Contract has no function to withdraw stuck ETH or ERC20 tokens. If ETH or ERC20 sent to contract by mistake, there wont be any ways to withdraw those funds.

Suggestion

Implement a function to be able to withdraw stuck ETH and ERC20 tokens from the contract (by owner)

FUNCTIONAL TESTING

Missing Logic – Immutable tax

Severity: Informational

Status: Open

Overview:

Fees are immutable and owner is not able to change fees later.

Suggestion

Its suggested to implement a function for updating fees depending on different market conditions in a safe range.

0 <= total buy fees <= 10

0 <= total sell fees <= 10

0 <= total transfer fees <= 10

https://docs.pinksale.finance/important/safu-contract

FUNCTIONAL TESTING

Missing Logic – Immutable internal swap threshold

Severity: Informational

Status: Open

Overview:

Internal swap threshold (swapTokensAtAmount) is immutable meaning that owner is not able to adjust this value depending on liquidity pool size or different market conditions.

Suggestion

Its suggested to implement a function for updating **swapTokensAtAmount** depending on liquidity pool size or different market conditions.

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