auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

$BABYPSYOP

DATED : 31 May 23'

AUDIT SUMMARY

Project name – $BABYPSYOP

Date: 31 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	1	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 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:
Contract has been tested on binance smart chain testnet which can be found in below link:
https://testnet.bscscan.com/token/0xB53EB0d36b3F255788FFEDE5da812BB231F64CAC

Token Information

Token Name : BabyPsyop

Token Symbol: $BABYPSYOP

Decimals: 18

Token Supply: 550,000,000,000

Token Address:

Checksum:
3eee40449e90d8132be4bcc9be51fe944a3e2724

Owner:
---(at time of writing the audit)

Deployer:

TOKEN OVERVIEW

Fees:

Buy Fees: 0-10%

Sell Fees: 0-10%

Transfer Fees: 0-5%

Fees Privilege: Owner

Ownership: Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: Changing swap threshold - changing fees - modifying swap settings - enabling trades

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

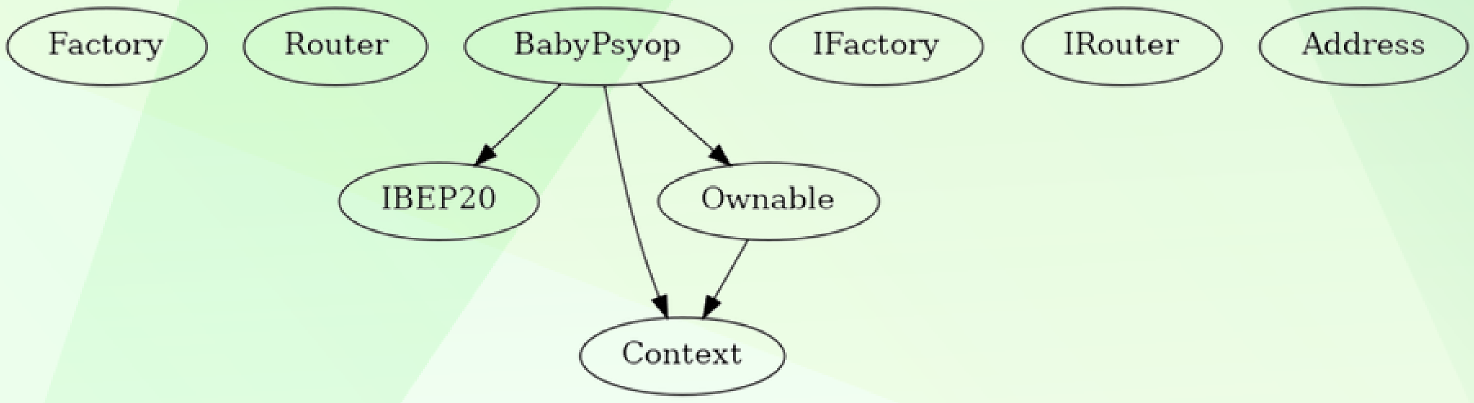
CLASSIFICATION OF RISK

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	1
🟨 Low-Risk	0
🟦 Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to change buy/sell fees over 10% and transfer fee over 5%
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to set max buy/sell/transfer/hold amount to 0
- Owner is not able to mint new tokens
- Owner must enable trades manually

CONTRACT ASSESMENT

```
| Contract |        Type        |       Bases      |                  |                 |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |  **Modifiers**  |
||||||
| **Factory** | Interface |  |||
| └ | createPair | External ❗️ | 🛑  |NO ❗️ |
||||||
| **Router** | Interface |  |||
| └ | WETH | External ❗️ |   |NO ❗️ |
| └ | factory | External ❗️ |   |NO ❗️ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗️ | 🛑  |NO ❗️ |
||||||
| **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 |
| └ | transferOwnership | 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 🔒 | 🛑  | |
||||||
| **BabyPsyop** | Implementation | Context, IBEP20, Ownable |||
```

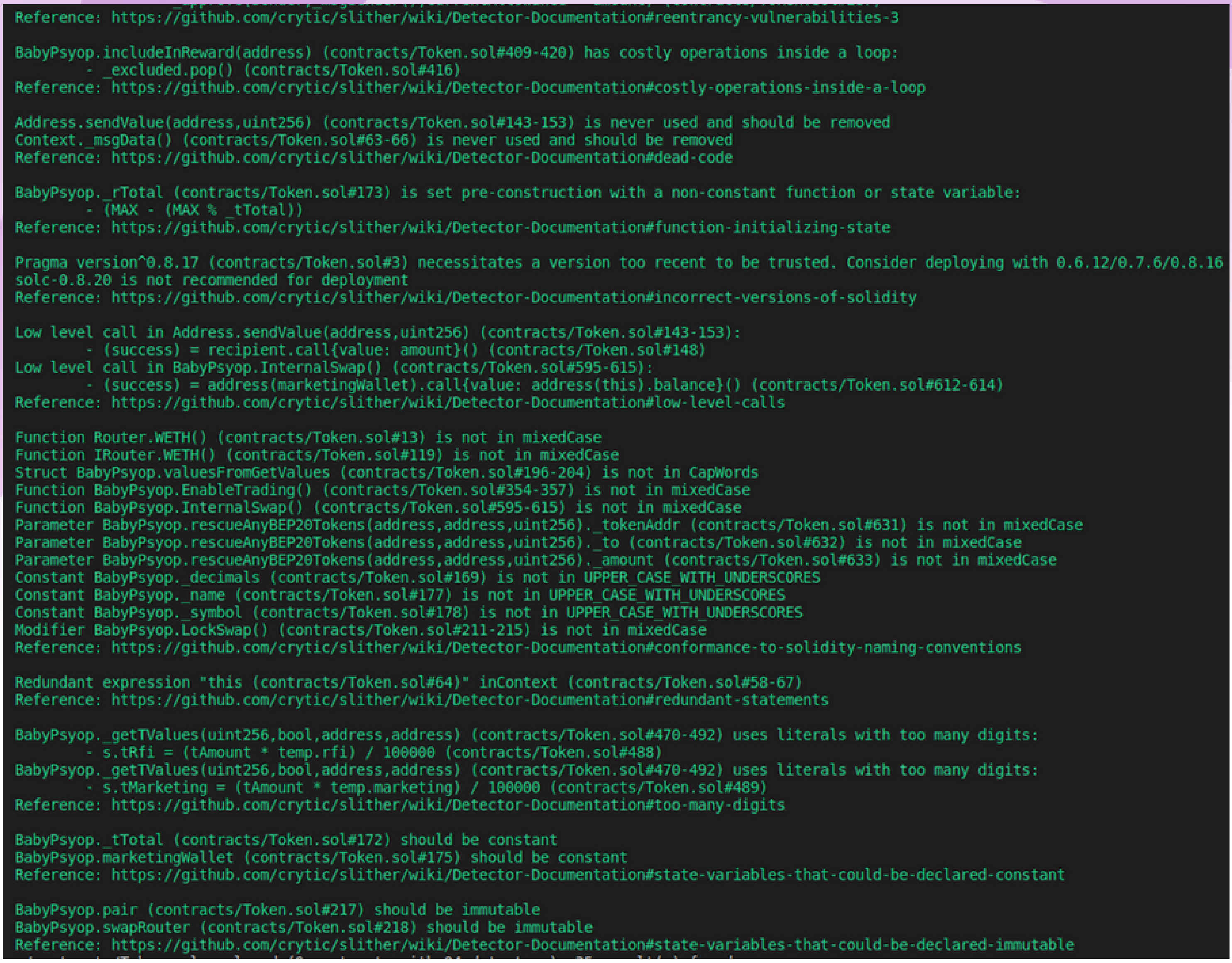
CONTRACT ASSESMENT

```
| └ | <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 ❗ |
| └ | EnableTrading | External ❗ | 🛑 | onlyOwner |
| └ | updateBuyTaxes | Public ❗ | 🛑 | onlyOwner |
| └ | updateSellTaxes | Public ❗ | 🛑 | onlyOwner |
| └ | updateTransferTaxes | Public ❗ | 🛑 | onlyOwner |
| └ | 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 🔐 | | |
| └ | _getRValues1 | Private 🔐 | | |
| └ | _getRate | Private 🔐 | | |
| └ | _getCurrentSupply | Private 🔐 | | |
| └ | _approve | Private 🔐 | 🛑 | |
| └ | _transfer | Private 🔐 | 🛑 | |
| └ | _tokenTransfer | Private 🔐 | 🛑 | |
| └ | InternalSwap | Internal 🔒 | 🛑 | LockSwap |
| └ | bulkExcludeFee | External ❗ | 🛑 | onlyOwner |
| └ | rescueBNB | External ❗ | 🛑 | onlyOwner |
| └ | rescueAnyBEP20Tokens | Public ❗ | 🛑 | onlyOwner |
| └ | <Receive Ether> | External ❗ | 💵 |NO ❗ |
```

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

BabyPsyop.includeInReward(address) (contracts/Token.sol#409-420) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#416)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Address.sendValue(address,uint256) (contracts/Token.sol#143-153) is never used and should be removed
Context._msgData() (contracts/Token.sol#63-66) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

BabyPsyop._rTotal (contracts/Token.sol#173) 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#3) 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#143-153):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#148)
Low level call in BabyPsyop.InternalSwap() (contracts/Token.sol#595-615):
        - (success) = address(marketingWallet).call{value: address(this).balance}() (contracts/Token.sol#612-614)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Router.WETH() (contracts/Token.sol#13) is not in mixedCase
Function IRouter.WETH() (contracts/Token.sol#119) is not in mixedCase
Struct BabyPsyop.valuesFromGetValues (contracts/Token.sol#196-204) is not in CapWords
Function BabyPsyop.EnableTrading() (contracts/Token.sol#354-357) is not in mixedCase
Function BabyPsyop.InternalSwap() (contracts/Token.sol#595-615) is not in mixedCase
Parameter BabyPsyop.rescueAnyBEP20Tokens(address,address,uint256)._tokenAddr (contracts/Token.sol#631) is not in mixedCase
Parameter BabyPsyop.rescueAnyBEP20Tokens(address,address,uint256)._to (contracts/Token.sol#632) is not in mixedCase
Parameter BabyPsyop.rescueAnyBEP20Tokens(address,address,uint256)._amount (contracts/Token.sol#633) is not in mixedCase
Constant BabyPsyop._decimals (contracts/Token.sol#169) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BabyPsyop._name (contracts/Token.sol#177) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BabyPsyop._symbol (contracts/Token.sol#178) is not in UPPER_CASE_WITH_UNDERSCORES
Modifier BabyPsyop.LockSwap() (contracts/Token.sol#211-215) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#64)" inContext (contracts/Token.sol#58-67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

BabyPsyop._getTValues(uint256,bool,address,address) (contracts/Token.sol#470-492) uses literals with too many digits:
        - s.tRfi = (tAmount * temp.rfi) / 100000 (contracts/Token.sol#488)
BabyPsyop._getTValues(uint256,bool,address,address) (contracts/Token.sol#470-492) uses literals with too many digits:
        - s.tMarketing = (tAmount * temp.marketing) / 100000 (contracts/Token.sol#489)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

BabyPsyop._tTotal (contracts/Token.sol#172) should be constant
BabyPsyop.marketingWallet (contracts/Token.sol#175) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

BabyPsyop.pair (contracts/Token.sol#217) should be immutable
BabyPsyop.swapRouter (contracts/Token.sol#218) 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

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity **(passed):**
https://testnet.bscscan.com/tx/0x33e3b8e63db8d4071b7a545f04260c30b89781601c7dbf51d4275553700e2e43

2- Buying when excluded (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0x69a9cd73d6c0c907e9fd4d8e544d3b385393a7aa6ee983ca059984738a42c2fc

3- Selling when excluded (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0x6d941777f39dd047399a9d13c829cc551936375218262dca9d5be0ea93d5de90

4- Transferring when excluded from fees (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0xbc9c930a377647f1f85170fba0e9f17ffa28999759f315a106a9a74bd01a6a0d

5- Buying when not excluded from fees (0-10% tax) **(passed):**
https://testnet.bscscan.com/tx/0x6d812df18f19441ca1e46bbd45e28952e61f27150ca7b8c65c982240d15c2b77

6- Selling when not excluded from fees (0-10% tax) **(passed):**
https://testnet.bscscan.com/tx/0xee08e3df61cb52df62720e9980b0779347231c3be607dcbfa2733c9a3387a2c0

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0-5% tax) **(passed):**
https://testnet.bscscan.com/tx/0x6da62a637a5d3a3622c82ff1c2a423339c841efe8fe6f38576ddb0acfa6fd9c8

8- Internal swap (marketing wallet received bnb) **(passed):**
https://testnet.bscscan.com/address/0x99a95a52953f1336f378047f00878ac0e100765f#internaltx

MANUAL TESTING

Centralization – Trades must be enabled

Severity: Medium

function: EnableTrading

Status: Not Resolved

Overview:

The smart contract owner must enable trades for holders. If trading remain disabled, no one would be able to buy/sell/transfer tokens.

```
function EnableTrading() external onlyOwner {
    require(!tradingEnabled, "Cannot re-enable trading");
    tradingEnabled = true;
}
```

Suggestion

To mitigate this centralization issue, we propose the following options:

1. Renounce Ownership: Consider relinquishing control of the smart contract by renouncing ownership. This would remove the ability for a single entity to manipulate the router, reducing centralization risks.
2. Multi-signature Wallet: Transfer ownership to a multi-signature wallet. This would require multiple approvals for any changes to the mainRouter, adding an additional layer of security and reducing the centralization risk.
3. Transfer ownership to a trusted and valid 3rd party in order to guarantee enabling of the trades

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