auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

BABYDOGEAI

DATED : 16 Dec 23'

AUDIT SUMMARY

Project name – BABYDOGEAI

Date: 16 Dec, 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	3	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/address/0x9e2e4ba4a54a0212ebb14491c5f92df6bb150c0c#readContract

Token Information

Token Address:
0x3eE8a21e8C146edDF9A73B68a7301291aCaB3a8F

Name: BABYDOGEAI

Symbol: BabyDogeAI

Decimals: 18

Network: EtherScan

Token Type: ERC-20

Owner: 0x89Eb21F253781C06BFDCE4d58CDb41c85f8c5439

Deployer:
0x89Eb21F253781C06BFDCE4d58CDb41c85f8c5439

Token Supply: 690000000000000000000000000000000

Checksum: 39bd5d4a707c73f24f6c3b6e8e0bb9a7

Testnet:
https://testnet.bscscan.com/address/0x9e2e4ba4a54a0212ebb14491c5f92df6bb150c0c#readContract

TOKEN OVERVIEW

Buy Fee: 0-10%

Sell Fee: 0-10%

Transfer Fee: 0-0%

Fee Privilege: Owner

Ownership: Owned

Minting: None

Max Tx: Yes

Blacklist: 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

- 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	3
Gas Optimization / Suggestions	1

INHERITANCE TREE

POINTS TO NOTE

- The owner can renounce the ownership.
- The owner can transfer ownership.
- The Owner can set the MKT address.
- The Owner can recuse tax.

STATIC ANALYSIS

```
INFO:Detectors:
BABYDOGEAI._transferToken(address,address,uint256,bool).taxFee (BabyDogeAI.sol#297) is a local variable never initialized
BABYDOGEAI._transferToken(address,address,uint256,bool).feeAmount (BabyDogeAI.sol#294) is a local variable never initialized
BABYDOGEAI._transfer(address,address,uint256).takeFee (BabyDogeAI.sol#263) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
BABYDOGEAI.constructor() (BabyDogeAI.sol#127-161) ignores return value by IERC20(_uniswapRouter.WETH()).approve(address(address(_uniswapRouter)),~ uint256(0
)) (BabyDogeAI.sol#156-159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
BABYDOGEAI.allowance(address,address).owner (BabyDogeAI.sol#196) shadows:
        - Ownable.owner() (BabyDogeAI.sol#85-87) (function)
BABYDOGEAI._approve(address,address,uint256).owner (BabyDogeAI.sol#213) shadows:
        - Ownable.owner() (BabyDogeAI.sol#85-87) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
BABYDOGEAI.recuseTax(uint256,uint256,uint256,uint256,uint256) (BabyDogeAI.sol#234-248) should emit an event for:
        - _finalBuyTax = newBuy (BabyDogeAI.sol#243)
        - _finalSellTax = newSell (BabyDogeAI.sol#244)
        - _reduceBuyTaxAt = newReduceBuy (BabyDogeAI.sol#245)
        - _reduceSellTaxAt = newReduceSell (BabyDogeAI.sol#246)
        - _preventSwapBefore = newPreventSwapBefore (BabyDogeAI.sol#247)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
BABYDOGEAI.setMKT(address,address).newMKT (BabyDogeAI.sol#164) lacks a zero-check on :
                - mkt = newMKT (BabyDogeAI.sol#167)
BABYDOGEAI.setMKT(address,address).newTeam (BabyDogeAI.sol#165) lacks a zero-check on :
                - team = newTeam (BabyDogeAI.sol#168)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in BABYDOGEAI.transferFrom(address,address,uint256) (BabyDogeAI.sol#205-211):
        External calls:
        - _transfer(sender,recipient,amount) (BabyDogeAI.sol#206)
                - _uniswapRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyDogeAI.sol#327-33
3)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (BabyDogeAI.sol#206)
                - mkt.transfer(_bal / 10) (BabyDogeAI.sol#337)
                - team.transfer(address(this).balance) (BabyDogeAI.sol#338)
        State variables written after the call(s):
```

```
INFO:Detectors:
Reentrancy in BABYDOGEAI._transfer(address,address,uint256) (BabyDogeAI.sol#250-285):
        External calls:
        - swapTokenForETH(_numSellToken) (BabyDogeAI.sol#271)
                - _uniswapRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyDogeAI.sol#327-33
3)
        External calls sending eth:
        - swapTokenForETH(_numSellToken) (BabyDogeAI.sol#271)
                - mkt.transfer(_bal / 10) (BabyDogeAI.sol#337)
                - team.transfer(address(this).balance) (BabyDogeAI.sol#338)
        Event emitted after the call(s):
        - Transfer(sender,address(this),swapAmount) (BabyDogeAI.sol#307)
                - _transferToken(from,to,amount,takeFee) (BabyDogeAI.sol#284)
        - Transfer(sender,recipient,tAmount - feeAmount) (BabyDogeAI.sol#312)
                - _transferToken(from,to,amount,takeFee) (BabyDogeAI.sol#284)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Pragma version^0.8.17 (BabyDogeAI.sol#20) allows old versions
solc-0.8.22 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Function IUniswapRouter.WETH() (BabyDogeAI.sol#41) is not in mixedCase
Parameter BABYDOGEAI.removeERC20(address)._token (BabyDogeAI.sol#316) is not in mixedCase
Variable BABYDOGEAI._isExcludeFromFee (BabyDogeAI.sol#115) is not in mixedCase
Variable BABYDOGEAI._uniswapRouter (BabyDogeAI.sol#117) is not in mixedCase
Variable BABYDOGEAI._uniswapPair (BabyDogeAI.sol#121) is not in mixedCase
Variable BABYDOGEAI._buyCount (BabyDogeAI.sol#225) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Reentrancy in BABYDOGEAI._transfer(address,address,uint256) (BabyDogeAI.sol#250-285):
        External calls:
        - swapTokenForETH(_numSellToken) (BabyDogeAI.sol#271)
                - mkt.transfer(_bal / 10) (BabyDogeAI.sol#337)
                - team.transfer(address(this).balance) (BabyDogeAI.sol#338)
        State variables written after the call(s):
        - _transferToken(from,to,amount,takeFee) (BabyDogeAI.sol#284)
                - _balances[sender] = _balances[sender] - tAmount (BabyDogeAI.sol#293)
                - _balances[address(this)] = _balances[address(this)] + swapAmount (BabyDogeAI.sol#306)
                - _balances[recipient] = _balances[recipient] + (tAmount - feeAmount) (BabyDogeAI.sol#311)
        - _buyCount ++ (BabyDogeAI.sol#280)
```

STATIC ANALYSIS

```
INFO:Detectors:
BABYDOGEAI.constructor() (BabyDogeAI.sol#127-161) uses literals with too many digits:
        - Supply = 69000000000000 (BabyDogeAI.sol#132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
BABYDOGEAI._initialBuyTax (BabyDogeAI.sol#226) should be constant
BABYDOGEAI._initialSellTax (BabyDogeAI.sol#227) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
BABYDOGEAI._decimals (BabyDogeAI.sol#114) should be immutable
BABYDOGEAI._totalSupply (BabyDogeAI.sol#116) should be immutable
BABYDOGEAI._uniswapPair (BabyDogeAI.sol#121) should be immutable
BABYDOGEAI._uniswapRouter (BabyDogeAI.sol#117) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:BabyDogeAI.sol analyzed (5 contracts with 93 detectors), 32 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

FUNCTIONAL TESTING

1- Approve (passed):

https://testnet.bscscan.com/tx/0x72e2ab28a856da8de2b752a0bfa1a5707382d835575c81ece25fe55381953099

2- Recuse Tax (passed):

https://testnet.bscscan.com/tx/0xdcb80f1909df1f4a791ec8acd67621c66cd8438783ad80ee977c199d70920dc3

3- Set Fee Exclude (passed):

https://testnet.bscscan.com/tx/0x8e31153fc336ae06c25f4ec958d8a71a7c8cf8b76d1d925c9ce9c751923e48d2

4- Set MKT (passed):

https://testnet.bscscan.com/tx/0x72b6319109eb27beef8300c39359eb6a9086d86fa0e061d666d1c974ba9c141d

MANUAL TESTING

Centralization – Local variable Shadowing

Severity: Low

Subject: Variable Shadowing

Status: Open

Overview:

```
function allowance (address owner, address spender) public view
override returns (uint256) {
    return _allowances[owner][spender];
 }

function approve (address spender, uint256 amount) public override
returns (bool) {
    _approve (msg.sender, spender, amount);
    return true;
 }
```

Suggestion:
Rename the local variables that shadow another component.

MANUAL TESTING

Centralization – Missing Events

Severity: Low

subject: Missing Events

Status: Open

Overview:

They serve as a mechanism for emitting and recording data onto the blockchain, making it transparent and easily accessible.

```
function setMKT(
    address payable newMKT,
    address payable newTeam
  ) public onlyOwner{
    mkt = newMKT;
    team = newTeam;
  }
function setFeeExclude(address account, bool value) public
onlyOwner{
    _isExcludeFromFee[account] = value;
  }
```

MANUAL TESTING

Centralization – Missing Zero Address

Severity: Low

Subject: Zero Check

Status: Open

Overview:

functions can take a zero address as a parameter (0x00000...). If a function parameter of address type is not properly validated by checking for zero addresses, there could be serious consequences for the contract's functionality.

```
function setMKT(
    address payable newMKT,
    address payable newTeam
  ) public onlyOwner{
    mkt = newMKT;
    team = newTeam;
  }
```

MANUAL TESTING

Optimization

Severity: Informational

subject: floating Pragma Solidity version

Status: Open

Overview:

It is considered best practice to pick one compiler version and stick with it. With a floating pragma, contracts may accidentally be deployed using an outdated.

pragma solidity ^0.8.17;

Suggestion:

Adding the latest constant version of solidity is recommended, as this prevents the unintentional deployment of a contract with an outdated compiler that contains unresolved bugs.

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