



Smart Contract Audit

FOR
BABYCAT INU

DATED : 9 March, 2024



MANUAL TESTING

Centralization – Missing Require Check

Severity: High

Function: Set Marketing Wallet

Status: Open

Overview:

The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner sets the address to the contract address, then the ETH will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.

```
function setMarketingWallet(address _marketingWallet) external onlyOwner {  
    require(_marketingWallet != address(0), "setmarketingWallet: ZERO");  
    marketingWallet = payable(_marketingWallet);  
    emit UpdateMarketingWallet(marketingWallet);  
}
```

Suggestion:

It is recommended that the address should not be able to be set as a contract address.



AUDIT SUMMARY

Project name - BABYCAT INU

Date: 9 March, 2024

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

Audit Status: Passed With High Risk

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	1	0	0	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/0x21f2db71415
afd2e148aa0a52b644c1b8a099432#code](https://testnet.bscscan.com/address/0x21f2db71415afd2e148aa0a52b644c1b8a099432#code)



Token Information

Token Name : BABYCAT INU

Token Symbol: BCAT

Decimals: 18

Token Supply: 420000000

Network: BscScan

Token Type: BEP-20

Token Address:

0x15A162c84D11008e054877c10d534310cd8B7fd

Checksum:

A2032c616934aeb47e6039f76b20d221

Owner:

0x8c66944F5247B8B9bc761F27c26C980601656079

(at time of writing the audit)

Deployer:

0x8c66944F5247B8B9bc761F27c26C980601656079



TOKEN OVERVIEW

Fees:

Buy Fee: 5%

Sell Fee: 5%

Fees Privilege: Owner

Ownership: Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: No

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 is exercised 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 Limit and 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

INHERITANCE TREE



STATIC ANALYSIS

A static analysis of the code was performed using Slither.
No issues were found.

```
INFO:Detectors:
BABYCATION.transferToAddressETH(address,uint256) (BABYCATION.sol#742-749) uses a dangerous strict equality:
  - amount == 0 (BABYCATION.sol#746)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
BABYCATION.constructor().currentRouter (BABYCATION.sol#548) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
BABYCATION.allowance(address,address)._owner (BABYCATION.sol#580) shadows:
  - Ownable._owner (BABYCATION.sol#194) (state variable)
BABYCATION._approve(address,address,uint256)._owner (BABYCATION.sol#628) shadows:
  - Ownable._owner (BABYCATION.sol#194) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Reentrancy in BABYCATION._transfer(address,address,uint256) (BABYCATION.sol#634-672):
  External calls:
    - swapAndLiquify() (BABYCATION.sol#654)
      - (succ) = recipient.call{value: amount}() (BABYCATION.sol#747)
      - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BABYCATION.sol#685-691)
  External calls sending eth:
    - swapAndLiquify() (BABYCATION.sol#654)
      - (succ) = recipient.call{value: amount}() (BABYCATION.sol#747)
  State variables written after the call(s):
    - marketingTokensCollected += fee (BABYCATION.sol#668)
    - totalMarketingTokensCollected += fee (BABYCATION.sol#669)
Reentrancy in BABYCATION.swapAndLiquify() (BABYCATION.sol#673-679):
  External calls:
    - swapTokensForEth(totalTokens) (BABYCATION.sol#675)
      - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BABYCATION.sol#685-691)
    - transferToAddressETH(marketingWallet,ethBalance) (BABYCATION.sol#677)
      - (succ) = recipient.call{value: amount}() (BABYCATION.sol#747)
  External calls sending eth:
    - transferToAddressETH(marketingWallet,ethBalance) (BABYCATION.sol#677)
      - (succ) = recipient.call{value: amount}() (BABYCATION.sol#747)
  State variables written after the call(s):
    - marketingTokensCollected = 0 (BABYCATION.sol#678)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
```

```
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (BABYCATION.sol#39-49):
  - (success) = recipient.call{value: amount}() (BABYCATION.sol#84)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (BABYCATION.sol#82-102):
  - (success,returnData) = target.call{value: value}(data) (BABYCATION.sol#92-94)
Low level call in Address.functionStaticCall(address,bytes,string) (BABYCATION.sol#114-127):
  - (success,returnData) = target.staticcall(data) (BABYCATION.sol#119)
Low level call in Address.functionDelegatecall(address,data,string) (BABYCATION.sol#139-152):
  - (success,returnData) = target.delegatecall(data) (BABYCATION.sol#144)
Low level call in BABYCATION.transferToAddressETH(address,uint256) (BABYCATION.sol#742-749):
  - (succ) = recipient.call{value: amount}() (BABYCATION.sol#747)
Low level call in BABYCATION.recoverETHFromContract() (BABYCATION.sol#766-772):
  - (succ) = address(marketingWallet).call{value: ethBalance}() (BABYCATION.sol#768)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (BABYCATION.sol#273) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (BABYCATION.sol#274) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (BABYCATION.sol#300) is not in mixedCase
Function IUniswapV2Router01.WETH() (BABYCATION.sol#326) is not in mixedCase
Parameter BABYCATION.allowance(address,address).._owner (BABYCATION.sol#580) is not in mixedCase
Parameter BABYCATION.setTokensToSwap(uint256).._minimumTokensBeforeSwap (BABYCATION.sol#723) is not in mixedCase
Parameter BABYCATION.setSwapAndLiquifyEnabled(bool).._enabled (BABYCATION.sol#732) is not in mixedCase
Parameter BABYCATION.setMarketingWallet(address).._marketingWallet (BABYCATION.sol#737) is not in mixedCase
Parameter BABYCATION.recoverTokensFromContract(address,uint256).._tokenAddress (BABYCATION.sol#774) is not in mixedCase
Parameter BABYCATION.recoverTokensFromContract(address,uint256).._amount (BABYCATION.sol#775) is not in mixedCase
Variable BABYCATION.WETH (BABYCATION.sol#538) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,address,uint256).amountADesired (BABYCATION.sol#330) is too similar to
  IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,address,uint256).amountBDesired (BABYCATION.sol#331)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
```

```
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (BABYCATION.sol#330) is too similar to
  IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (BABYCATION.sol#331)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
BABYCATION._tTotal (BABYCATION.sol#526) should be constant
BABYCATION.buyFee (BABYCATION.sol#530) should be constant
BABYCATION.sellFee (BABYCATION.sol#532) should be constant
BABYCATION.transferFee (BABYCATION.sol#531) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Slither:BABYCATION.sol analyzed (9 contracts with 93 detectors), 59 result(s) found
```



FUNCTIONAL TESTING

1- Approve (passed):

<https://testnet.bscscan.com/tx/0xc78dc98e3603a626c8c25bb207d15457cc481afc6922f0062b1eae9762a1b26a>

2- Increase Allowance (passed):

<https://testnet.bscscan.com/tx/0x823b0da5c5ec0fb71df1adb8a90b3c5384bf76e020eacfc8eda66daf57f7b50>

3- Decrease Allowance (passed):

<https://testnet.bscscan.com/tx/0x1a3c4b77568deb0c49256df4da1e228ed179b4388082370052da099e877462f5>

4- Exclude From Fee (passed):

<https://testnet.bscscan.com/tx/0x028fa377c22186137d06bfb0f692432c13ce4ece54656a90287ae22695324e19>

5- Set Marketing Wallet (passed):

<https://testnet.bscscan.com/tx/0x9442da56f8362f31f43257edf9e305210293ba429f5184f2cb3f20f3e78dc0aa>

6- Transfer (passed):

<https://testnet.bscscan.com/tx/0xd842d834e92ce8137554742dbbd1aa5d7f6eeac51dd677af9e7de8db9957329>



POINTS TO NOTE

- The owner can transfer ownership.
- The owner can renounce ownership.
- The owner can set the marketing wallet address.
- The owner can exclude/include address from fees.
- The owner can set tokens to swap.



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	1
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	1



MANUAL TESTING

Centralization – Missing Require Check

Severity: High

Function: Set Marketing Wallet

Status: Open

Overview:

The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner sets the address to the contract address, then the ETH will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.

```
function setMarketingWallet(address _marketingWallet) external onlyOwner {  
    require(_marketingWallet != address(0), "setmarketingWallet: ZERO");  
    marketingWallet = payable(_marketingWallet);  
    emit UpdateMarketingWallet(marketingWallet);  
}
```

Suggestion:

It is recommended that the address should not be able to be set as a contract address.



MANUAL TESTING

Optimization

Severity: Optimization

Function: Remove unused code.

Status: Open

Overview:

Unused variables are allowed in Solidity, and they do. not pose a direct security issue. It is the best practice. though to avoid them.

```
event AuditLog(string, address);
event UpdateStakingWallet(address);
event UpdateTransferFee(uint256);
event UpdateDistribution(uint256, uint256);
event TradingStarted(bool);
function _msgData() internal view virtual returns (bytes calldata) {
    return msg.data;
}
function sendValue(address payable recipient, uint256 amount) internal {
    require(
        address(this).balance >= amount,
        "Address: insufficient balance"
    );
    (bool success, ) = recipient.call{value: amount}("");
    require(
        success,
        "Address: unable to send value, recipient may have reverted"
    );
}
function functionCall(
    address target,
    bytes memory data
) internal returns (bytes memory) {
    return
        functionCallWithValue(
            target,
```



MANUAL TESTING

```
        data,
        0,
        "Address: low-level call failed"
    );
}
function functionCall(
    address target,
    bytes memory data,
    string memory errorMessage
) internal returns (bytes memory) {
    return functionCallWithValue(target, data, 0, errorMessage);
}
function functionCallWithValue(
    address target,
    bytes memory data,
    uint256 value
) internal returns (bytes memory) {
    return
        functionCallWithValue(
            target,
            data,
            value,
            "Address: low-level call with value failed"
        );
}
function functionStaticCall(
    address target,
    bytes memory data
) internal view returns (bytes memory) {
    return
        functionStaticCall(
            target,
            data,
            "Address: low-level static call failed"
        );
}
```



MANUAL TESTING

```
function functionDelegateCall(
    address target,
    bytes memory data
) internal returns (bytes memory) {
    return
        functionDelegateCall(
            target,
            data,
            "Address: low-level delegate call failed"
        );
}
interface IUniswapV2Pair {
    event Approval(
        address indexed owner,
        address indexed spender,
        uint256 value
    );
}
```



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 specialize 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>
