AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Astro Grok

DATED : 26 Dec 23'

AUDIT SUMMARY

Project name – Astro Grok

Date: 26 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	2	2
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/0xfbbd7a2904373f7c5dfb0f4a38d8ef52dde838a9#code

Token Information

Token Address:
0x1b5d82313fbd6F6D4757C17993719bddBE690eD1

Name: Astro Grok

Symbol: AGRK

Decimals: 18

Network: BscScan

Token Type: BEP-20

Owner: 0xd8FF642952d6bBe728821d5f9fA8287A27CD3795

Deployer:
0xd8FF642952d6bBe728821d5f9fA8287A27CD3795

Token Supply: 1000000000

Checksum: 89032c616934aeb47e6039f76b20d2e5

Testnet:
https://testnet.bscscan.com/address/0xfbbd7a2904373f7c5dfb0f4a38d8ef52dde838a9#code

TOKEN OVERVIEW

Buy Fee: 0-0%

Sell Fee: 0-0%

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

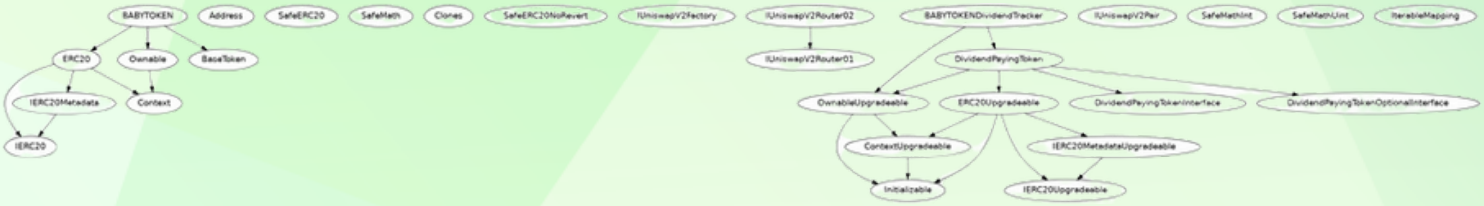
CLASSIFICATION OF RISK

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	2
Gas Optimization / Suggestions	2

INHERITANCE TREE



POINTS TO NOTE

- The owner can transfer ownership.
- The owner can renounce ownership.
- The owner can exclude wallet addresses from dividends.
- The owner can set a Balance.
- The owner can set swap tokens at amount.
- The owner can exclude wallets from fees.
- The owner can update gas for processing.
- The owner can update the marketing wallet address.

STATIC ANALYSIS

```
INFO:Detectors:
BABYTOKEN.getAccountDividendsInfo(address) (BabyToken.sol#3215-3230) ignores return value by dividendTracker.getAccount(account) (BabyToken.sol#3229)
BABYTOKEN.getAccountDividendsInfoAtIndex(uint256) (BabyToken.sol#3232-3247) ignores return value by dividendTracker.getAccountAtIndex(index) (BabyToken.sol#3246)
BABYTOKEN.claim() (BabyToken.sol#3265-3267) ignores return value by dividendTracker.processAccount(address(msg.sender),false) (BabyToken.sol#3266)
BABYTOKEN.addLiquidity(uint256,uint256) (BabyToken.sol#3438-3451) ignores return value by uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyT
oken.sol#3443-3450)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
DividendPayingToken.__DividendPayingToken_init(address,string,string)._name (BabyToken.sol#2463) shadows:
        - ERC20Upgradeable._name (BabyToken.sol#1728) (state variable)
DividendPayingToken.__DividendPayingToken_init(address,string,string)._symbol (BabyToken.sol#2464) shadows:
        - ERC20Upgradeable._symbol (BabyToken.sol#1729) (state variable)
DividendPayingToken.dividendOf(address)._owner (BabyToken.sol#2524) shadows:
        - OwnableUpgradeable._owner (BabyToken.sol#2072) (state variable)
DividendPayingToken.withdrawableDividendOf(address)._owner (BabyToken.sol#2531) shadows:
        - OwnableUpgradeable._owner (BabyToken.sol#2072) (state variable)
DividendPayingToken.withdrawnDividendOf(address)._owner (BabyToken.sol#2543) shadows:
        - OwnableUpgradeable._owner (BabyToken.sol#2072) (state variable)
DividendPayingToken.accumulativeDividendOf(address)._owner (BabyToken.sol#2557) shadows:
        - OwnableUpgradeable._owner (BabyToken.sol#2072) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
BABYTOKEN.setSwapTokensAtAmount(uint256) (BabyToken.sol#3073-3079) should emit an event for:
        - swapTokensAtAmount = amount (BabyToken.sol#3078)
BABYTOKEN.setTokenRewardsFee(uint256) (BabyToken.sol#3111-3115) should emit an event for:
        - totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee) (BabyToken.sol#3113)
BABYTOKEN.setLiquiditFee(uint256) (BabyToken.sol#3117-3121) should emit an event for:
        - liquidityFee = value (BabyToken.sol#3118)
        - totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee) (BabyToken.sol#3119)
BABYTOKEN.setMarketingFee(uint256) (BabyToken.sol#3123-3127) should emit an event for:
        - marketingFee = value (BabyToken.sol#3124)
        - totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee) (BabyToken.sol#3125)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
BABYTOKEN.constructor(string,string,uint256,address[4],uint256[3],uint256,address,uint256)._uniswapV2Pair (BabyToken.sol#3044-3045) lacks a zero-check on :
            - uniswapV2Pair = _uniswapV2Pair (BabyToken.sol#3047)
BABYTOKEN.constructor(string,string,uint256,address[4],uint256[3],uint256,address,uint256).serviceFeeReceiver_ (BabyToken.sol#3010) lacks a zero-check on :
            - address(serviceFeeReceiver_).transfer(serviceFee_) (BabyToken.sol#3068)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
SafeERC20NoRevert.safeTransfer(IERC20,address,uint256) (BabyToken.sol#1227-1239) has external calls inside a loop: (success,returndata) = address(token).call(abi.encodeWithSelector(token.transfer.selector,to,v
alue)) (BabyToken.sol#1232-1234)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop
INFO:Detectors:
Reentrancy in BABYTOKEN._transfer(address,address,uint256) (BabyToken.sol#3277-3365):
        External calls:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
            - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (BabyToken.sol#816)
```

```
INFO:Detectors:
Reentrancy in BABYTOKEN._transfer(address,address,uint256) (BabyToken.sol#3277-3365):
        External calls:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
            - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (BabyToken.sol#816)
            - IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance) (BabyToken.sol#3376)
            - (success,returndata) = target.call{value: value}(data) (BabyToken.sol#636)
            - uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3429-3435)
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
            - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyToken.sol#3443-3450)
            - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3411-3417)
        External calls sending eth:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
            - (success,returndata) = target.call{value: value}(data) (BabyToken.sol#636)
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
            - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyToken.sol#3443-3450)
        State variables written after the call(s):
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
            - _allowances[owner][spender] = amount (BabyToken.sol#459)
Reentrancy in BABYTOKEN._transfer(address,address,uint256) (BabyToken.sol#3277-3365):
        External calls:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
            - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (BabyToken.sol#816)
            - IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance) (BabyToken.sol#3376)
            - (success,returndata) = target.call{value: value}(data) (BabyToken.sol#636)
            - uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3429-3435)
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
            - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyToken.sol#3443-3450)
            - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3411-3417)
        - swapAndSendDividends(sellTokens) (BabyToken.sol#3320)
            - (success,returndata) = address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value)) (BabyToken.sol#1232-1234)
            - success = SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),address(dividendTracker),dividends) (BabyToken.sol#3456-3460)
            - dividendTracker.distributeCAKEDividends(dividends) (BabyToken.sol#3462)
            - uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3429-3435)
        External calls sending eth:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
            - (success,returndata) = target.call{value: value}(data) (BabyToken.sol#636)
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
            - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyToken.sol#3443-3450)
        State variables written after the call(s):
        - swapAndSendDividends(sellTokens) (BabyToken.sol#3320)
            - _allowances[owner][spender] = amount (BabyToken.sol#459)
Reentrancy in BABYTOKENDividendTracker.processAccount(address,bool) (BabyToken.sol#2879-2893):
        External calls:
        - amount = _withdrawDividendOfUser(account) (BabyToken.sol#2884)
            - (success,returndata) = address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value)) (BabyToken.sol#1232-1234)
            - success = SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),user,_withdrawableDividend) (BabyToken.sol#2503-2507)
        State variables written after the call(s):
        - lastClaimTimes[account] = block.timestamp (BabyToken.sol#2887)
Reentrancy in BABYTOKEN.swapAndLiquify(uint256) (BabyToken.sol#3379-3400):
```

STATIC ANALYSIS

```
INFO:Detectors:
Reentrancy in BABYTOKEN._setAutomatedMarketMakerPair(address,bool) (BabyToken.sol#3129-3141):
        External calls:
        - dividendTracker.excludeFromDividends(pair) (BabyToken.sol#3137)
        Event emitted after the call(s):
        - SetAutomatedMarketMakerPair(pair,value) (BabyToken.sol#3140)
Reentrancy in BABYTOKEN._transfer(address,address,uint256) (BabyToken.sol#3277-3365):
        External calls:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (BabyToken.sol#816)
                - IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance) (BabyToken.sol#3376)
                - (success,returndata) = target.call{value: value}(data) (BabyToken.sol#636)
                - uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3429-3435)
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyToken.sol#3443-3450)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3411-3417)
        External calls sending eth:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
                - (success,returndata) = target.call{value: value}(data) (BabyToken.sol#636)
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyToken.sol#3443-3450)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (BabyToken.sol#460)
                - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
        - SwapAndLiquify(half,newBalance,otherHalf) (BabyToken.sol#3399)
                - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
Reentrancy in BABYTOKEN._transfer(address,address,uint256) (BabyToken.sol#3277-3365):
        External calls:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (BabyToken.sol#816)
                - IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance) (BabyToken.sol#3376)
                - (success,returndata) = target.call{value: value}(data) (BabyToken.sol#636)
                - uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3429-3435)
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyToken.sol#3443-3450)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3411-3417)
        - swapAndSendDividends(sellTokens) (BabyToken.sol#3320)
                - (success,returndata) = address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value)) (BabyToken.sol#1232-1234)
                - success = SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),address(dividendTracker),dividends) (BabyToken.sol#3456-3460)
                - dividendTracker.distributeCAKEDividends(dividends) (BabyToken.sol#3462)
                - uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (BabyToken.sol#3429-3435)
        External calls sending eth:
        - swapAndSendToFee(marketingTokens) (BabyToken.sol#3308)
                - (success,returndata) = target.call{value: value}(data) (BabyToken.sol#636)
        - swapAndLiquify(swapTokens) (BabyToken.sol#3315)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,address(0xdead),block.timestamp) (BabyToken.sol#3443-3450)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (BabyToken.sol#460)
```

```
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (BabyToken.sol#1290) is too similar to IUniswapV2Router01.addLiquidity(address,address,u
int256,uint256,uint256,uint256,address,uint256).amountBDesired (BabyToken.sol#1291)
Variable DividendPayingToken.__DividendPayingToken_init(address,string,string)._rewardToken (BabyToken.sol#2462) is too similar to BABYTOKENDividendTracker.initialize(address,uint256).rewardToken_ (BabyToken.s
ol#2655)
Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend (BabyToken.sol#2496) is too similar to BABYTOKENDividendTracker.getAccount(address).withdrawableDividends (BabyToken.sol#2735
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
Clones.clone(address) (BabyToken.sol#1152-1161) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_clone_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d730000000000000000000000000000000000000000) (BabyToken.sol#1155)
Clones.clone(address) (BabyToken.sol#1152-1161) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_clone_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf3000000000000000000000000000000000000000000) (BabyToken.sol#1157)
Clones.cloneDeterministic(address,bytes32) (BabyToken.sol#1170-1179) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_cloneDeterministic_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d730000000000000000000000000000000000000000) (BabyToken.sol#1173)
Clones.cloneDeterministic(address,bytes32) (BabyToken.sol#1170-1179) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_cloneDeterministic_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf3000000000000000000000000000000000000000000) (BabyToken.sol#1175)
Clones.predictDeterministicAddress(address,bytes32,address) (BabyToken.sol#1184-1199) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_predictDeterministicAddress_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d730000000000000000000000000000000000000000) (BabyToken.sol#1191)
Clones.predictDeterministicAddress(address,bytes32,address) (BabyToken.sol#1184-1199) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_predictDeterministicAddress_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf3ff0000000000000000000000000000000000000000) (BabyToken.sol#1193)
BABYTOKEN.constructor(string,string,uint256,address[4],uint256[3],uint256,address,uint256) (BabyToken.sol#3003-3069) uses literals with too many digits:
        - gasForProcessing = 300000 (BabyToken.sol#3032)
BABYTOKEN.updateGasForProcessing(uint256) (BabyToken.sol#3143-3154) uses literals with too many digits:
        - require(bool,string)(newValue >= 200000 && newValue <= 500000,BABYTOKEN: gasForProcessing must be between 200,000 and 500,000) (BabyToken.sol#3144-3147)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
SafeMathInt.MAX_INT256 (BabyToken.sol#2197) is never used in SafeMathInt (BabyToken.sol#2195-2252)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
INFO:Detectors:
BABYTOKEN.dividendTracker (BabyToken.sol#2954) should be immutable
BABYTOKEN.rewardToken (BabyToken.sol#2956) should be immutable
BABYTOKEN.uniswapV2Pair (BabyToken.sol#2958) should be immutable
BABYTOKEN.uniswapV2Router (BabyToken.sol#2949) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:BabyToken.sol analyzed (29 contracts with 93 detectors), 120 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/0xbc1b96394d7dbd2db3a077b47c608348c84d53e55ed5df65a4510e726e57f870

2- Increase Allowance (passed):

https://testnet.bscscan.com/tx/0x691782ff4f544eddf1584dda1f8e8b0344007023c82df3bb29c573d2d5909379

3- Decrease Allowance (passed):

https://testnet.bscscan.com/tx/0x5d696b9a3472ab9c449a53e1959bfb95ea98c9bb109f2122974b5812d9192e01

4- Exclude From Dividends (passed):

https://testnet.bscscan.com/tx/0x59ec51cb33d9156990b24f55f6b25ed81cde29ca5cd5bbbcfacf080e85483921

5- Transfer Ownership (passed):

https://testnet.bscscan.com/tx/0x5b128399d385c82c7195823cf9f2aac3433c75c0ffd784ce6997a9bd9ce0c039

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 setSwapTokensAtAmount(uint256 amount) external
onlyOwner {
require(
     amount > totalSupply() / 10**5,
"BABYTOKEN: Amount must be greater than 0.001% of total
supply"
   );
   swapTokensAtAmount = amount;
}
 function setMarketingWallet(address payable wallet) external
onlyOwner {
require(
     wallet != address(0),
"BABYTOKEN: The marketing wallet cannot be the value of zero"
   );
require(!wallet.contract(), "Marketing wallet cannot be a
contract");
   _marketingWalletAddress = wallet;
 }
```

MANUAL TESTING

```
function setTokenRewardsFee(uint256 value) external onlyOwner {
    tokenRewardsFee = value;
    totalFees =
tokenRewardsFee.add(liquidityFee).add(marketingFee);
require(totalFees <= 25, "Total fee is over 25%");
  }
function setLiquiditFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees =
tokenRewardsFee.add(liquidityFee).add(marketingFee);
require(totalFees <= 25, "Total fee is over 25%");
  }
 function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees =
tokenRewardsFee.add(liquidityFee).add(marketingFee);
require(totalFees <= 25, "Total fee is over 25%");
  }
```

Suggestion:
Emit an event for critical changes.

MANUAL TESTING

Centralization – Missing Zero Address

Severity: Low

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.

```
constructor(
    string memory name_,
    string memory symbol_,
    uint256 totalSupply_,
    address[4] memory addrs, // reward, router, marketing wallet,
dividendTracker
    uint256[3] memory feeSettings, // rewards, liquidity, marketing
    uint256 minimumTokenBalanceForDividends_,
    address serviceFeeReceiver_,
    uint256 serviceFee_
  ) payable ERC20(name_, symbol_) {
    rewardToken = addrs[0];
    _marketingWalletAddress = addrs[2];
    require(
      msg.sender != _marketingWalletAddress,
      "Owner and marketing wallet cannot be the same"
    );
    require(
      !_marketingWalletAddress.isContract(),
      "Marketing wallet cannot be a contract"
    );
```

MANUAL TESTING

```
    tokenRewardsFee = feeSettings[0];
    liquidityFee = feeSettings[1];
    marketingFee = feeSettings[2];
    totalFees =
tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
    swapTokensAtAmount = totalSupply_.div(1000); // 0.1%

    // use by default 300,000 gas to process auto-claiming
dividends
    gasForProcessing = 300000;

    dividendTracker = BABYTOKENDividendTracker(
      payable(Clones.clone(addrs[3]))
    );
    dividendTracker.initialize(
      rewardToken,
      minimumTokenBalanceForDividends_
    );
```

Suggestion:
It is suggested that the address should not be zero or dead.

MANUAL TESTING

Optimization

Severity: Optimization

subject: 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

```
  function _msgData() internal view virtual returns (bytes calldata)
{
    return msg.data;
  }
}
```

MANUAL TESTING

Optimization

Severity: Informational

subject: Remove Safe Math

Status: Open

Line: 914-1125

Overview:

compiler version above 0.8.0 can control arithmetic overflow/underflow, it is recommended to remove the unwanted code to avoid high gas fees.

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