AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

PEPETATE

DATED : 23 MAY 23'

AUDIT SUMMARY

Project name – PEPETATE

Date: 23 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: The code has undergone a line-by-line review by the **Ace** team.

2.ETH Test Network: All tests were conducted on the ETH 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/0xDB74C0ca0 3cA178c820e6848B6eED1DdE0eC78b5#code

Token Information

Name : Pepe Tate

Symbol : PEPETATE

Decimals: 18

Network: BSC

Token Type:BEP20

Token Address:
0x297ea5EbE431D0802F5888181EB7546989Eacc11

Owner:
0x50C32d22d0561bcFd8c2fE064eb259Ae3438115F
(at time of writing the audit)

Deployer:0x50C32d22d0561bcFd8c2fE064eb259Ae3438115F

Token Information

Fees:

Buy Fees: 0-25%

Sell Fees: 0-25%

Transfer Fees: 0-25%

Fees Privilige: Owner

Ownership : Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: - Modifying fees

- including in rewards

- excluding from rewards

- changing swap threshold

- changing reward settings (min token to be eligible and cooldown)

- 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
- ✅ **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
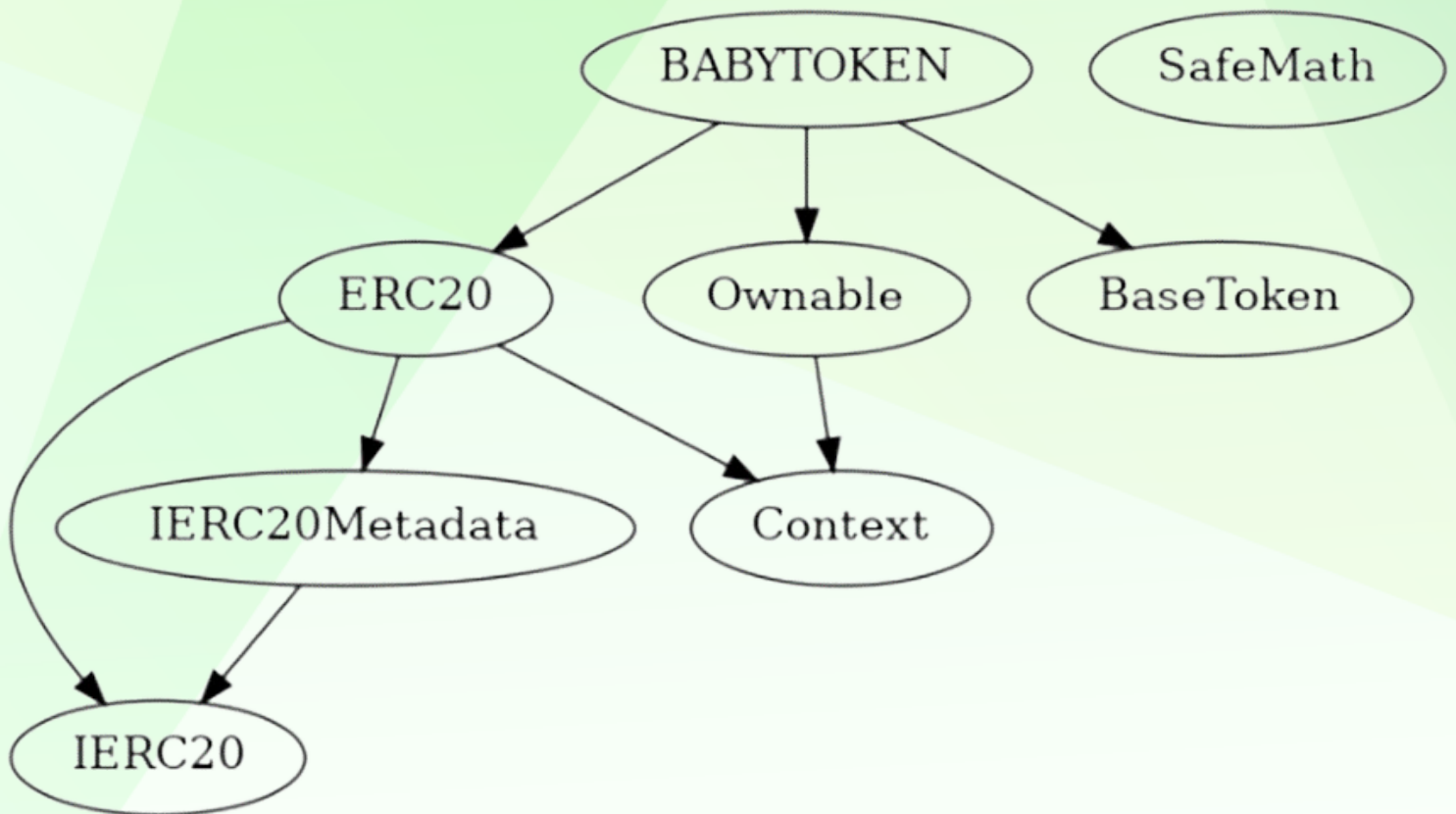
POINTS TO NOTE

- Owner is not able to change buy/sell/transfer fees within range 0-25%
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to limit buy/sell/transfer/wallet amounts
- Owner is not able to mint new tokens
- **Token is pinksale generated (already tested) and uses $BTC as reward token.**

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
└	totalSupply	External ❗	|NO ❗	
└	balanceOf	External ❗	|NO ❗	
└	transfer	External ❗	🔴 |NO ❗	
└	allowance	External ❗	|NO ❗	
└	approve	External ❗	🔴 |NO ❗	
└	transferFrom	External ❗	🔴 |NO ❗	
IERC20Metadata	Interface	IERC20		
└	name	External ❗	|NO ❗	
└	symbol	External ❗	|NO ❗	
└	decimals	External ❗	|NO ❗	
Context	Implementation			
└	_msgSender	Internal 🔐		
└	_msgData	Internal 🔐		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	<Constructor>	Public ❗	🔴 |NO ❗	
└	name	Public ❗	|NO ❗	
└	symbol	Public ❗	|NO ❗	
└	decimals	Public ❗	|NO ❗	
└	totalSupply	Public ❗	|NO ❗	
└	balanceOf	Public ❗	|NO ❗	
└	transfer	Public ❗	🔴 |NO ❗	
└	allowance	Public ❗	|NO ❗	
└	approve	Public ❗	🔴 |NO ❗	
└	transferFrom	Public ❗	🔴 |NO ❗	
└	increaseAllowance	Public ❗	🔴 |NO ❗	
└	decreaseAllowance	Public ❗	🔴 |NO ❗	
└	_transfer	Internal 🔐	🔴	
└	_mint	Internal 🔐	🔴	
└	_burn	Internal 🔐	🔴	
└	_approve	Internal 🔐	🔴	
└	_beforeTokenTransfer	Internal 🔐	🔴	
└	_afterTokenTransfer	Internal 🔐	🔴	
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🔴 |NO ❗	

CONTRACT ASSESMENT

└	owner	Public ❗		NO ❗
└	renounceOwnership	Public ❗	🔴	onlyOwner
└	transferOwnership	Public ❗	🔴	onlyOwner
└	_setOwner	Private 🔐	🔴	
SafeMath	Library			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
Clones	Library			
└	clone	Internal 🔒	🔴	
└	cloneDeterministic	Internal 🔒	🔴	
└	predictDeterministicAddress	Internal 🔒		
└	predictDeterministicAddress	Internal 🔒		
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🔴	
└	functionCall	Internal 🔒	🔴	
└	functionCall	Internal 🔒	🔴	
└	functionCallWithValue	Internal 🔒	🔴	
└	functionCallWithValue	Internal 🔒	🔴	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🔴	
└	functionDelegateCall	Internal 🔒	🔴	
└	verifyCallResult	Internal 🔒		
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO ❗
└	feeToSetter	External ❗		NO ❗

CONTRACT ASSESMENT

└	getPair	External ❗		NO ❗
└	allPairs	External ❗		NO ❗
└	allPairsLength	External ❗		NO ❗
└	createPair	External ❗	🔴	NO ❗
└	setFeeTo	External ❗	🔴	NO ❗
└	setFeeToSetter	External ❗	🔴	NO ❗
IUniswapV2Router01	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🔴	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	removeLiquidity	External ❗	🔴	NO ❗
└	removeLiquidityETH	External ❗	🔴	NO ❗
└	removeLiquidityWithPermit	External ❗	🔴	NO ❗
└	removeLiquidityETHWithPermit	External ❗	🔴	NO ❗
└	swapExactTokensForTokens	External ❗	🔴	NO ❗
└	swapTokensForExactTokens	External ❗	🔴	NO ❗
└	swapExactETHForTokens	External ❗	💵	NO ❗
└	swapTokensForExactETH	External ❗	🔴	NO ❗
└	swapExactTokensForETH	External ❗	🔴	NO ❗
└	swapETHForExactTokens	External ❗	💵	NO ❗
└	quote	External ❗		NO ❗
└	getAmountOut	External ❗		NO ❗
└	getAmountIn	External ❗		NO ❗
└	getAmountsOut	External ❗		NO ❗
└	getAmountsIn	External ❗		NO ❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
IERC20Upgradeable	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🔴	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🔴	NO ❗
└	transferFrom	External ❗	🔴	NO ❗

CONTRACT ASSESMENT

IERC20MetadataUpgradeable	Interface	IERC20Upgradeable		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
Initializable	Implementation			
ContextUpgradeable	Implementation	Initializable		
└	__Context_init	Internal 🔒	🔴	initializer
└	__Context_init_unchained	Internal 🔒	🔴	initializer
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgradeable		
└	__ERC20_init	Internal 🔒	🔴	initializer
└	__ERC20_init_unchained	Internal 🔒	🔴	initializer
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🔴	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🔴	NO❗
└	transferFrom	Public ❗	🔴	NO❗
└	increaseAllowance	Public ❗	🔴	NO❗
└	decreaseAllowance	Public ❗	🔴	NO❗
└	_transfer	Internal 🔒	🔴	
└	_mint	Internal 🔒	🔴	
└	_burn	Internal 🔒	🔴	
└	_approve	Internal 🔒	🔴	
└	_beforeTokenTransfer	Internal 🔒	🔴	
└	_afterTokenTransfer	Internal 🔒	🔴	
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
└	__Ownable_init	Internal 🔒	🔴	initializer
└	__Ownable_init_unchained	Internal 🔒	🔴	initializer
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🔴	onlyOwner
└	transferOwnership	Public ❗	🔴	onlyOwner

CONTRACT ASSESMENT

└	_setOwner	Private 🔐	🔴	
IUniswapV2Pair	Interface			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🔴	NO❗
└	transfer	External ❗	🔴	NO❗
└	transferFrom	External ❗	🔴	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🔴	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	mint	External ❗	🔴	NO❗
└	burn	External ❗	🔴	NO❗
└	swap	External ❗	🔴	NO❗
└	skim	External ❗	🔴	NO❗
└	sync	External ❗	🔴	NO❗
└	initialize	External ❗	🔴	NO❗
SafeMathInt	Library			
└	mul	Internal 🔐	
└	div	Internal 🔐	
└	sub	Internal 🔐	
└	add	Internal 🔐	
└	abs	Internal 🔐	
└	toUint256Safe	Internal 🔐	
SafeMathUint	Library			
└	toInt256Safe	Internal 🔐	

CONTRACT ASSESMENT

IterableMapping	Library				
└	get	Internal 🔒			
└	getIndexOfKey	Internal 🔒			
└	getKeyAtIndex	Internal 🔒			
└	size	Internal 🔒			
└	set	Internal 🔒	🔴		
└	remove	Internal 🔒	🔴		
DividendPayingTokenInterface	Interface			
└	dividendOf	External ❗			NO ❗
└	withdrawDividend	External ❗	🔴	NO ❗
DividendPayingTokenOptionalInterface	Interface			
└	withdrawableDividendOf	External ❗			NO ❗
└	withdrawnDividendOf	External ❗			NO ❗
└	accumulativeDividendOf	External ❗			NO ❗
DividendPayingToken	Implementation	ERC20Upgradeable, OwnableUpgradeable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
└	__DividendPayingToken_init	Internal 🔒	🔴	initializer
└	distributeCAKEDividends	Public ❗	🔴	onlyOwner
└	withdrawDividend	Public ❗	🔴	NO ❗
└	_withdrawDividendOfUser	Internal 🔒	🔴		
└	dividendOf	Public ❗			NO ❗
└	withdrawableDividendOf	Public ❗			NO ❗
└	withdrawnDividendOf	Public ❗			NO ❗
└	accumulativeDividendOf	Public ❗			NO ❗
└	_transfer	Internal 🔒	🔴		
└	_mint	Internal 🔒	🔴		
└	_burn	Internal 🔒	🔴		
└	_setBalance	Internal 🔒	🔴		
BABYTOKENDividendTracker	Implementation	OwnableUpgradeable, DividendPayingToken		
└	initialize	External ❗	🔴	initializer
└	_transfer	Internal 🔒			
└	withdrawDividend	Public ❗		NO ❗
└	excludeFromDividends	External ❗	🔴	onlyOwner
└	isExcludedFromDividends	Public ❗		NO ❗
└	updateClaimWait	External ❗	🔴	onlyOwner
└	updateMinimumTokenBalanceForDividends	External ❗	🔴	onlyOwner
└	getLastProcessedIndex	External ❗		NO ❗
└	getNumberOfTokenHolders	External ❗		NO ❗

CONTRACT ASSESMENT

└	getAccount	Public ❗		NO ❗
└	getAccountAtIndex	Public ❗		NO ❗
└	canAutoClaim	Private 🔐		
└	setBalance	External ❗	🔴	onlyOwner
└	process	Public ❗	🔴	NO ❗
└	processAccount	Public ❗	🔴	onlyOwner
BaseToken	Implementation			
BABYTOKEN	Implementation	ERC20, Ownable, BaseToken		
└	<Constructor>	Public ❗	💵	ERC20
└	<Receive Ether>	External ❗	💵	NO ❗
└	setSwapTokensAtAmount	External ❗	🔴	onlyOwner
└	excludeFromFees	External ❗	🔴	onlyOwner
└	excludeMultipleAccountsFromFees	External ❗	🔴	onlyOwner
└	setMarketingWallet	External ❗	🔴	onlyOwner
└	setTokenRewardsFee	External ❗	🔴	onlyOwner
└	setLiquiditFee	External ❗	🔴	onlyOwner
└	setMarketingFee	External ❗	🔴	onlyOwner
└	_setAutomatedMarketMakerPair	Private 🔐	🔴	
└	updateGasForProcessing	Public ❗	🔴	onlyOwner
└	updateClaimWait	External ❗	🔴	onlyOwner
└	getClaimWait	External ❗		NO ❗
└	updateMinimumTokenBalanceForDividends	External ❗	🔴	onlyOwner
└	getMinimumTokenBalanceForDividends	External ❗		NO ❗
└	getTotalDividendsDistributed	External ❗		NO ❗
└	isExcludedFromFees	Public ❗		NO ❗
└	withdrawableDividendOf	Public ❗		NO ❗
└	dividendTokenBalanceOf	Public ❗		NO ❗
└	excludeFromDividends	External ❗	🔴	onlyOwner
└	isExcludedFromDividends	Public ❗		NO ❗
└	getAccountDividendsInfo	External ❗		NO ❗
└	getAccountDividendsInfoAtIndex	External ❗		NO ❗
└	processDividendTracker	External ❗	🔴	NO ❗
└	claim	External ❗	🔴	NO ❗
└	getLastProcessedIndex	External ❗		NO ❗
└	getNumberOfDividendTokenHolders	External ❗		NO ❗
└	_transfer	Internal 🔒	🔴	
└	swapAndSendToFee	Private 🔐	🔴	
└	swapAndLiquify	Private 🔐	🔴	
└	swapTokensForEth	Private 🔐	🔴	

CONTRACT ASSESMENT

└	swapTokensForCake	Private 🔐	🔴	
└	addLiquidity	Private 🔐	🔴	
└	swapAndSendDividends	Private 🔐	🔴	

Legend

Symbol	Meaning
🔴	Function can modify state
💲	Function is payable

STATIC ANALYSIS

```
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#2209) is not in mixedCase
Function DividendPayingToken.__DividendPayingToken_init(address,string,string) (contracts/Token.sol#2507-2515) is not in mixedCase
Parameter DividendPayingToken.__DividendPayingToken_init(address,string,string)._rewardToken (contracts/Token.sol#2508) is not in mixedCase
Parameter DividendPayingToken.__DividendPayingToken_init(address,string,string)._name (contracts/Token.sol#2509) is not in mixedCase
Parameter DividendPayingToken.__DividendPayingToken_init(address,string,string)._symbol (contracts/Token.sol#2510) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address)._owner (contracts/Token.sol#2568) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address)._owner (contracts/Token.sol#2576) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address)._owner (contracts/Token.sol#2585) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address)._owner (contracts/Token.sol#2596) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/Token.sol#2487) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter BABYTOKENdividendTracker.getAccount(address)._account (contracts/Token.sol#2757) is not in mixedCase
Variable BABYTOKEN._marketingWalletAddress (contracts/Token.sol#2982) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#1261) is too simi
lar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#1262)
Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend (contracts/Token.sol#2541) is too similar to BABYTOKENdividendTracker.getAc
count(address).withdrawableDividends (contracts/Token.sol#2765)
Variable DividendPayingToken.__DividendPayingToken_init(address,string,string)._rewardToken (contracts/Token.sol#2508) is too similar to BABYTOKENdividendTrack
er.initialize(address,uint256).rewardToken_ (contracts/Token.sol#2690)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

Clones.clone(address) (contracts/Token.sol#880-895) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_clone_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d73000000000000000000000000) (contracts/Token.sol#883-886)
Clones.clone(address) (contracts/Token.sol#880-895) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_clone_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf30000000000000000000000000000000000) (contracts/Token.sol#888-891)
Clones.cloneDeterministic(address,bytes32) (contracts/Token.sol#904-922) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_cloneDeterministic_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d73000000000000000000000000) (contracts/Token.sol#910-913
)
Clones.cloneDeterministic(address,bytes32) (contracts/Token.sol#904-922) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_cloneDeterministic_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf30000000000000000000000000000000000) (contracts/Token.sol#
915-918)
Clones.predictDeterministicAddress(address,bytes32,address) (contracts/Token.sol#927-948) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_predictDeterministicAddress_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d73000000000000000000000000) (contracts/Token.so
l#934-937)
Clones.predictDeterministicAddress(address,bytes32,address) (contracts/Token.sol#927-948) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr_predictDeterministicAddress_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf3ff0000000000000000000000000000000000) (contracts/T
oken.sol#939-942)
BABYTOKEN.constructor(string,string,uint256,address[4],uint256[3],uint256) (contracts/Token.sol#3020-3082) uses literals with too many digits:
        - gasForProcessing = 300000 (contracts/Token.sol#3047)
BABYTOKEN.updateGasForProcessing(uint256) (contracts/Token.sol#3155-3166) uses literals with too many digits:
        - require(bool,string)(newValue >= 200000 && newValue <= 500000,BABYTOKEN: gasForProcessing must be between 200,000 and 500,000) (contracts/Token.sol#3
156-3159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

SafeMathInt.MAX_INT256 (contracts/Token.sol#2256) is never used in SafeMathInt (contracts/Token.sol#2254-2311)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

BABYTOKEN.dividendTracker (contracts/Token.sol#2971) should be immutable
BABYTOKEN.rewardToken (contracts/Token.sol#2973) should be immutable
BABYTOKEN.uniswapV2Pair (contracts/Token.sol#2967) should be immutable
BABYTOKEN.uniswapV2Router (contracts/Token.sol#2966) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Static Analysis

an static analysis of the code were performed using slither. No issues were found

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0xa392663e37ba25dc9835a6700 d9ddc50a9e5d378209e27ff388c2fdca64df00e

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x13b2f1cf4a032c21d7752b493a7 971a32c72b9716424e0635ae36b115509ab64

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xddc1c08b588225ed98b529536d 70bfbd8f299517685b6c0d0c52ac72b2db8722

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xa76aa9b4ee2fce5a4c4ecd3967 63e3150480836b43dd950b82965d79249a3ddf

5- Buying from a regular wallet (0-25% tax) (passed):
https://testnet.bscscan.com/tx/0x4f96baa531cb2e617a5128e1df0 2ac4a2562c6ac7f4c88f7102218dc14595673

6- Selling from a regular wallet (0-25% tax) (passed):
https://testnet.bscscan.com/tx/0x29b6bf50e7ace568f22a8b6423 4b62297a59f52c59ca8848c2f86c744e93d119

7- Transferring from a regular wallet (0-25% tax) (passed):
https://testnet.bscscan.com/tx/0xa037f475b41a21c35d4c1065265 5e734ebd9b652c44725dbc6c9e9dbdebdd221

FUNCTIONAL TESTING

7- Internal swap + rewards distirbution + auto liquidity(passed):
https://testnet.bscscan.com/tx/0x29b6bf50e7ace568f22a8b6423
4b62297a59f52c59ca8848c2f86c744e93d119

FUNCTIONAL TESTING

Category: Centralization
Subject: Excessive fees
Severity: Medium
Overview:
Owner is able to set buy/sell/transfer each one up to 25%

Code:
function setTokenRewardsFee(uint256 value) external onlyOwner { ... }
function setLiquiditFee(uint256 value) external onlyOwner { ... }
function setMarketingFee(uint256 value) external onlyOwner { ... }

Suggestion:
Implement a limit for max amount of buy/sell/transfer fees.
Buy Fees <= 10
Sell Fees <= 10
Transfer Fees <= 10

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