SAFUAUDIT
SMART CONTRACT AUDITING

# BETCOIN
## SMART CONTRACT AUDIT


March 22, 2022

# INTRODUCTION

| | |
|---|---|
| **Client** | Betcoin (BET) |
| **Language** | Solidity |
| **Contract Address** | 0xaA187b877697c56Ea85D6E4192058c756749De92 |
| **Decimals** | 18 |
| **Supply** | 1,000,000,000 |
| **Platform** | Binance Smart Chain |
| **Compiler** | v0.8.4+commit.c7e474f2 |
| **Optimization** | Yes, with 200 runs |
| **Website** | https://betcoinbsc.com/ |
| **Telegram** | https://t.me/BetcoinBSC |
| **Twitter** | https://twitter.com/BetcoinBsc |



# Description

Betting has been made fun, risk free and helping Betcoin price increase with Pegged Bitcoin Reflections. Each sports bet we will create 2 New wallets one for team A the other for Team B. Winners will receive 1.5x to 2x their bet no more than 2 hours after event ends, tokens from loosing wallet will be burned helping price increase with every bet and incorporating Live Casino Games in Q3.

# **TABLE** OF CONTENTS

# Approach

## Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.

## Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch, that lead to scams and rugpulls.

,

## Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
- Back-doors
- Vulnerability
- Accuracy
- Readability

## Tools

- Remix IDE
- MythX, Mythril
- SWC Registry
- Open Zeppelin Code Analyzer
- Solidity Code Complier

# RISK CLASSIFICATION

## CRITICAL

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

## MEDIUM

Issues on this level could potentially bring problems and should eventually be fixed.

## MINOR

Issues on this level are minor  details and warning that can remain unfixed but would be better fixed at some point in the future

## INFORMATIONAL

Information level is to offer suggestions for improvement of efficacity or security for features  with a risk free factor.

# ABSTRACT

| Fees | Ownership | Mint function |
| --- | --- | --- |
| Buy Fees: 11%<br>Sell Fees: 11%<br>*at audit time | Owned | No mint function found |

| Owner can set fees | Max Tx amount | Pause |
| --- | --- | --- |
| Owner can't set fees over 25% | Owner can't set max Tx amount | Owner can't pause trading |

# Vulnerabilities Test

| SWC ID | Description | |
|---|---|---|
| SWC-100 | Function Default Visibility | Passed |
| SWC-101 | Integer Overflow and Underflow | Passed |
| SWC-102 | Outdated Compiler Version | Passed |
| SWC-103 | Floating Pragma | Passed |
| SWC-104 | Unchecked Call Return Value | Passed |
| SWC-105 | Unprotected Ether Withdrawal | Passed |
| SWC-106 | Unprotected SELF-DESTRUCT Instruction | Passed |
| SWC-107 | Re-entrancy | Passed |
| SWC-108 | State Variable Default Visibility | Passed |
| SWC-109 | Uninitialized Storage Pointer | Passed |
| SWC-110 | Assert Violation | Passed |
| SWC-111 | Use of Deprecated Solidity Functions | Passed |
| SWC-112 | Delegate Call to Untrusted Callee | Passed |
| SWC-113 | DoS with Failed Call | Passed |
| SWC-114 | Transaction Order Dependence | Passed |
| SWC-115 | Authorization through tx.origin | Minor |

| | | |
|---|---|---|
| **SWC-116** | Block values as a proxy for time | **Passed** |
| **SWC-117** | Signature Malleability | **Passed** |
| **SWC-118** | Incorrect Constructor Name | **Passed** |
| **SWC-119** | Shadowing State Variables | **Passed** |
| **SWC-120** | Weak Sources of Randomness from Chain Attributes | **Passed** |
| **SWC-121** | Missing Protection against Signature Replay Attacks | **Passed** |
| **SWC-122** | Lack of Proper Signature Verification | **Passed** |
| **SWC-123** | Requirement Violation | **Passed** |
| **SWC-124** | Write to Arbitrary Storage Location | **Passed** |
| **SWC-125** | Incorrect Inheritance Order | **Passed** |
| **SWC-126** | Insufficient Gas Griefing | **Passed** |
| **SWC-127** | Arbitrary Jump with Function Type Variable | **Passed** |
| **SWC-128** | DoS With Block Gas Limit | **Passed** |
| **SWC-129** | Typographical Error | **Passed** |
| **SWC-130** | Right-To-Left-Override control character (U+202E) | **Passed** |
| **SWC-131** | Presence of unused variables | **Passed** |
| **SWC-132** | Unexpected Ether balance | **Passed** |
| **SWC-133** | Hash Collisions With Multiple Variable Length Arguments | **Passed** |
| **SWC-134** | Message call with the hardcoded gas amount | **Passed** |
| **SWC-135** | Code With No Effects (Irrelevant/Dead Code) | **Passed** |
| **SWC-136** | Unencrypted Private Data On-Chain | **Passed** |

# MANUAL ANALYSIS

The contract is verified to check if functions do and work as they should and malicious code is not inserted.

| | Tested | Result |
|---|---|---|
| **Transfer** | Yes | **Passed** |
| **Total Supply** | Yes | **Passed** |
| **Buy Back** | Yes | **N/A** |
| **Burn** | Yes | **N/A** |
| **Mint** | Yes | **N/A** |
| **Rebase** | Yes | **N/A** |
| **Pause** | Yes | **N/A** |
| **Blacklist** | Yes | **N/A** |
| **Lock** | Yes | **N/A** |
| **Max Transaction** | Yes | **N/A** |
| **Transfer Ownership** | Yes | **Passed** |
| **Renounce Ownership** | Yes | **Passed** |

# CONTRACT INSPECTION

| **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❗ |
| └ | 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 🔒 | | |
||||||
| **IUniswapV2Factory** | Interface | |||
| └ | feeTo | External ❗ | |NO❗ |
| └ | feeToSetter | External ❗ | |NO❗ |
| └ | 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❗ |
||||||
| **IERC20MetadataUpgradeable** | Interface | IERC20Upgradeable |||
| └ | name | External ❗ | |NO❗ |
| └ | symbol | 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 |
| └ | _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 🔒 | | |

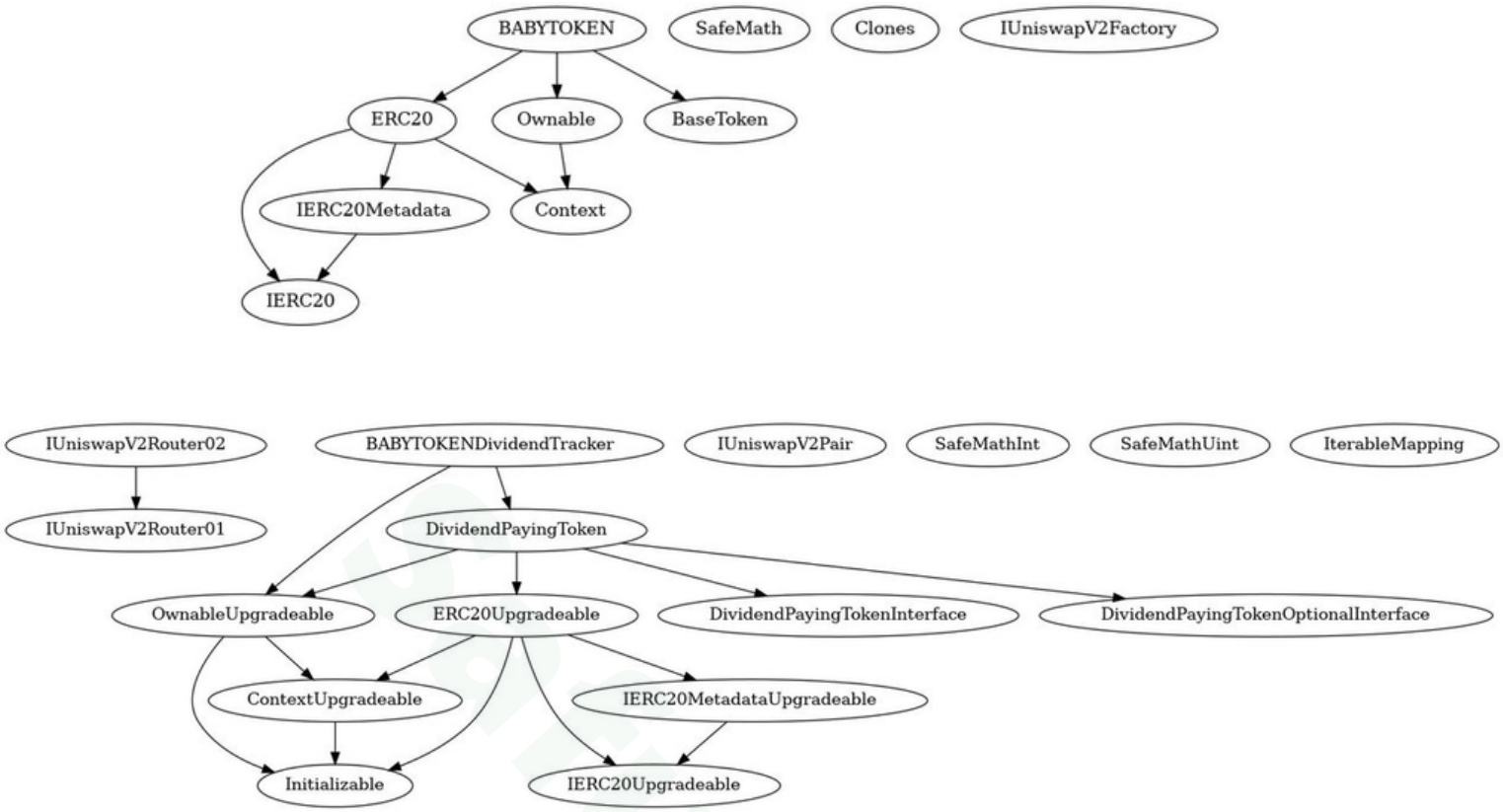
| **IterableMapping** | Library | |||
| └ | get | Public ❗ | |NO❗ |
| └ | getIndexOfKey | Public ❗ | |NO❗ |
| └ | getKeyAtIndex | Public ❗ | |NO❗ |
| └ | size | Public ❗ | |NO❗ |
| └ | set | Public ❗ | 🛑 |NO❗ |
| └ | remove | Public ❗ | 🛑 |NO❗ |
||||||
| **DividendPayingTokenInterface** | Interface | |||
| └ | 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❗ |
| └ | 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 |
| └ | updateDividendTracker | Public ❗ | 🛑 | onlyOwner |
| └ | updateUniswapV2Router | Public ❗ | 🛑 | onlyOwner |
| └ | excludeFromFees | Public ❗ | 🛑 | onlyOwner |
| └ | excludeMultipleAccountsFromFees | Public ❗ | 🛑 | onlyOwner |
| └ | setMarketingWallet | External ❗ | 🛑 | onlyOwner |
| └ | setTokenRewardsFee | External ❗ | 🛑 | onlyOwner |
| └ | setLiquiditFee | External ❗ | 🛑 | onlyOwner |
| └ | setMarketingFee | External ❗ | 🛑 | onlyOwner |
| └ | setAutomatedMarketMakerPair | Public ❗ | 🛑 | 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 🔐 | 🛑 | |
| └ | swapTokensForCake | Private 🔐 | 🛑 | |
| └ | addLiquidity | Private 🔐 | 🛑 | |
| └ | swapAndSendDividends | Private 🔐 | 🛑 | |

| Symbol | Meaning |
|---|---|
| 🛑 | Function can modify state |
| 💵 | Function is payable |
| 🔐 | Private function |
| 🔒 | Internal function |
| NO❗ | Function has no modifier |

# INHERITANCE TREE




Inheritance is a feature of the object-oriented programming language. It is a way of extending the functionality of a program, used to separate the code, reduces the dependency, and increases the re-usability of the existing code. Solidity supports inheritance between smart contracts, where multiple contracts can be inherited into a single contract.

# Important Snippets

### Owner can't set buy/sell fees over 25%

```
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%");
}
```

### Owner can exclude multiple accounts from fees

```
function excludeMultipleAccountsFromFees(
    address[] calldata accounts,
    bool excluded
) public onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

### Owner can exclude from dividends

```
function excludeFromDividends(address account) external onlyOwner {
    require(!excludedFromDividends[account]);
    excludedFromDividends[account] = true;

    _setBalance(account, 0);
    tokenHoldersMap.remove(account);

    emit ExcludeFromDividends(account);
}
```

# GOOD PRACTICES

- The owner cannot mint new tokens
- The owner cannot set max Tx amount
- The owner cannot stop or pause the smart contract
- The owner cannot set fees over 25%
- The smart contract utilizes "SafeMath" to prevent overflows

```
library SafeMath {

    function add(uint256 a, uint256 b) internal pure returns (uint256) {
        uint256 c = a + b;
        require(c >= a, "SafeMath: addition overflow");

        return c;
    }

    function sub(uint256 a, uint256 b) internal pure returns (uint256) {
        return sub(a, b, "SafeMath: subtraction overflow");
    }

    function sub(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
        require(b <= a, errorMessage);
        uint256 c = a - b;

        return c;
    }

    function mul(uint256 a, uint256 b) internal pure returns (uint256) {
        // Gas optimization: this is cheaper than requiring 'a' not being zero, but the
        // benefit is lost if 'b' is also tested.
        // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522
        if (a == 0) {
            return 0;
        }

        uint256 c = a * b;
        require(c / a == b, "SafeMath: multiplication overflow");

        return c;
    }

    function div(uint256 a, uint256 b) internal pure returns (uint256) {
        return div(a, b, "SafeMath: division by zero");
    }

    function div(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
        require(b > 0, errorMessage);
        uint256 c = a / b;
        // assert(a == b * c + a % b); // There is no case in which this doesn't hold

        return c;
    }

    function mod(uint256 a, uint256 b) internal pure returns (uint256) {
        return mod(a, b, "SafeMath: modulo by zero");
    }

    function mod(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
        require(b != 0, errorMessage);
        return a % b;
    }
}
```

# WEBSITE

| | |
|---|---|
| **Website** | https://betcoinbsc.com/ |
| **Domain Registry** | https://www.godaddy.com |
| **Domain Expiry Date** | 2023-03-20 |
| **Response Code** | 200 |
| **SSL Checker and HTTPS Test** | Passed |
| **Deprecated HTML tags** | Passed |
| **Robots.txt** | Passed |
| **Sitemap Test** | Passed |
| **SEO Friendly URL** | Passed |
| **Responsive Test** | Passed |
| **JS Error Test** | Passed |
| **Console Errors Test** | Passed |
| **Site Loading Speed Test** | 0.97 seconds - Passed |
| **HTTP2 Test** | Passed |
| **Safe Browsing Test** | Passed |

# DISCLAIMER

SafuAudit.com is not a financial institution and the information provided on this website does not constitute investment advice, financial advice, trading advice or any other sort of advice. You should not treat any of the website's content as such. Investing in crypto assets carries a high level of risk and does not hold guarantees for not sustaining financial loss due to their volatility.

Accuracy of Information

SafuAudit will strive to ensure accuracy of information listed on this website although it will not hold any responsibility for any missing or wrong information. SafuAudit provides all information as is. You understand that you are using any and all information available here at your own risk. Any use or reliance on our content and services is solely at your own risk and discretion.

The purpose of the audit is to analyse the on-chain smart contract source code, and to provide basic overview of the project.

While we have used all the information available to us for this straightforward investigation, you should not rely on this report only — we recommend proceeding with several independent audits Be aware that smart contracts deployed on a blockchain aren't secured enough against external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, SafuAudit does not guarantee the explicit security of the audited smart contract. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

# AUDIT RESULTS

## CRITICAL

No critical severity issues have been found.

## MEDIUM

No medium severity issues have been found.

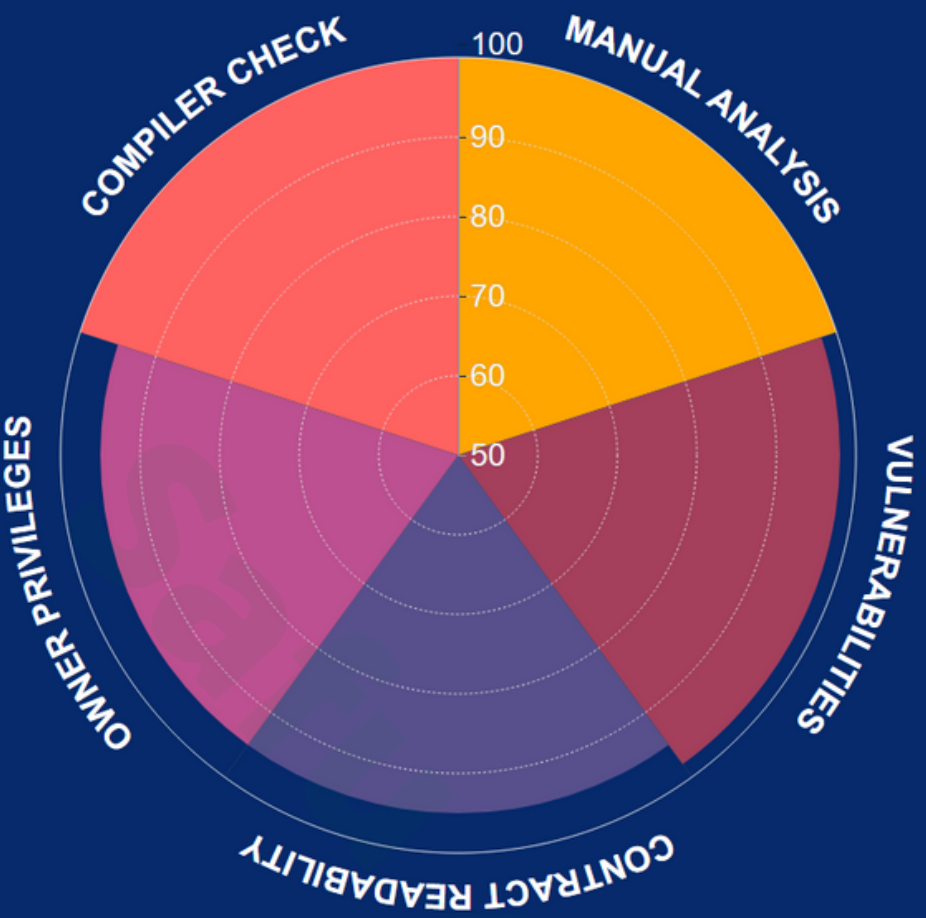
## MINOR

- Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" instead.

## INFORMATIONAL

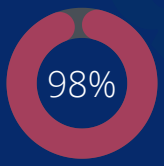
The standard audit model does not offer suggestions and consulting for improvements of efficacity.

# SCORE



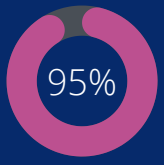


Manual Analysis




Vulnerabilities


Contract Readability



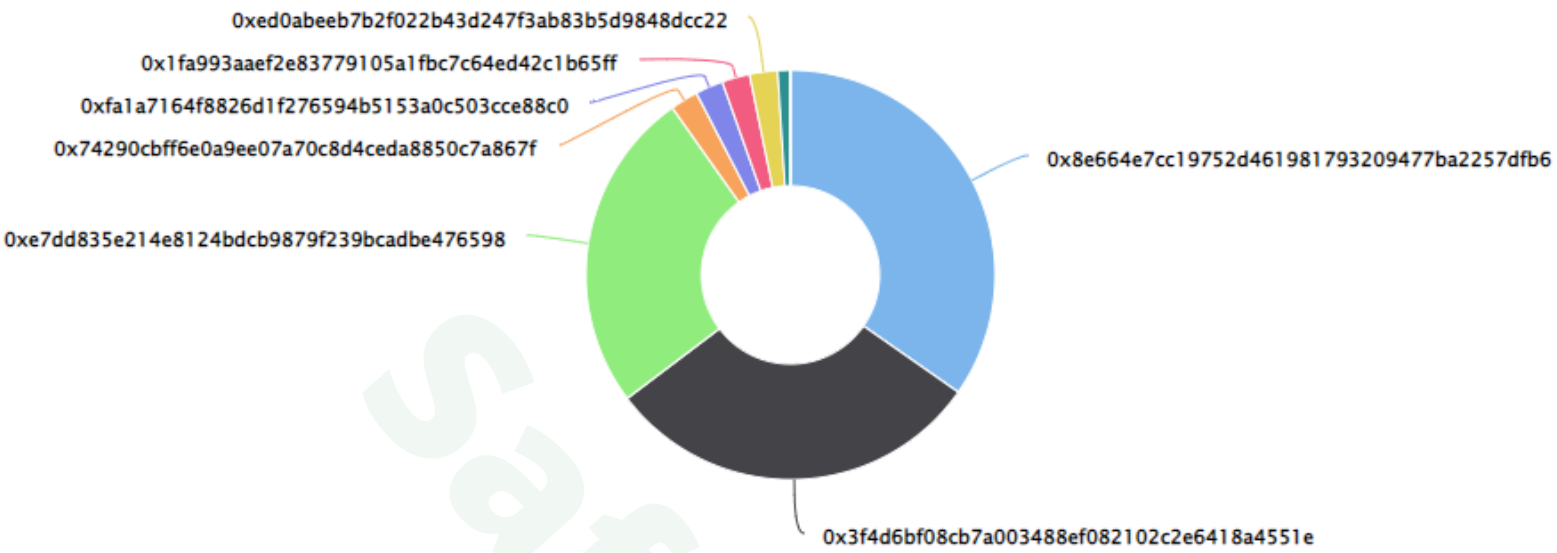

Owner Privileges


Compiler Check

## Final Score:97.6

# SUMMARY

## Top 10 holders




| Rank | Address | Quantity (Token) | Percentage |
|---|---|---|---|
| 1 | 0x8e664e7cc19752d461981793209477ba2257dfb6 | 347,100,000 | 34.7100% |
| 2 | 0x3f4d6bf08cb7a003488ef082102c2e6418a4551e | 300,000,000 | 30.0000% |
| 3 | 0xe7dd835e214e8124bdcb9879f239bcadbe476598 | 254,900,000 | 25.4900% |
| 4 | 0x74290cbff6e0a9ee07a70c8d4ceda8850c7a867f | 22,000,000 | 2.2000% |
| 5 | 0xfa1a7164f8826d1f276594b5153a0c503cce88c0 | 22,000,000 | 2.2000% |
| 6 | 0x1fa993aaef2e83779105a1fbc7c64ed42c1b65ff | 22,000,000 | 2.2000% |
| 7 | 0xed0abeeb7b2f022b43d247f3ab83b5d9848dcc22 | 22,000,000 | 2.2000% |
| 8 | 0x1cbe174661178ff5d01d110000f1ba10a69819ed | 10,000,000 | 1.0000% |



# Conclusion

Project Betcoin does not contain any severe issues or risk characteristics.

SafuAudit has tested the security based on manual and automated tests.

Please note that we don't offer any warranties for business model.

SafuAudit.com