Betscoin (BETC) Smart Contract Documentation

Overview

Betscoin (BETC) is an ERC20 token implemented on the Binance Smart Chain (BSC). The contract includes features for presale management, staking, liquidity management, token distribution, tax management, and security measures. It integrates OpenZeppelin's standard libraries for ERC20 functionality, ownership control, reentrancy protection, and pausability.

Key Features

1. Token Details

- **Name**: Betscoin
- **Symbol**: BETC
- **Total Supply**: 10,000,000,000 BETC (10 billion)
- **Decimals**: 18

2. Tokenomics

- **Presale Allocation**: 20% (2,000,000,000 BETC)
- **Liquidity Allocation**: 10% (1,000,000,000 BETC)
- **Staking Allocation**: 15% (1,500,000,000 BETC)
- **Team and Advisors Allocation**: 11.87% (1,187,000,000 BETC)
- **Ecosystem Allocation**: 29.44% (2,944,000,000 BETC)
- **Marketing Allocation**: 7% (700,000,000 BETC)
- **Airdrop Allocation**: 6.69% (669,000,000 BETC)
- **Total Distributed Tokens**: A new field to track the total number of tokens distributed during presale phases and other distributions.

3. Presale Management

- **Phases**: 8 presale phases, each with different token amounts, prices, and maximum purchase limits per user.
- **Functionality**: Users can participate in the ongoing presale through the participateInPresale() function, which ensures tokens are allocated according to the current phase.
- **Referral Bonus**: A 10% referral bonus is granted to referrers during presale participation if the phase has sufficient tokens.
- **End Presale**: The owner can end the presale at any time using the endPresale() function.

4. Staking System

- **Staking Periods**: Users can stake tokens for 6, 12, 24, or 48 months.
- **Annual Interest Rates**:
  - 6 months: 29%
  - 12 months: 51%
  - 24 months: 60%
  - 48 months: 73%
- **Rewards**: Staking rewards are calculated and minted when claimed using the claimStakedRewards() function.

5. Liquidity Management

- **Pair Creation**: A function to create a PancakeSwap pair and add liquidity using the createPairAndAddLiquidity() function.
- **Liquidity Lock**: Liquidity added is locked for 365 days, and the owner can remove liquidity after the lock period using the removeLiquidity() function.

6. Token Distribution

• **Team & Advisors Vesting**: Team and advisors' tokens are distributed according to a **12-month cliff** and **36-month linear vesting**. The distributeTeamAdvisorTokens() function handles the distribution.

• **Ecosystem Tokens**: Ecosystem tokens are locked for 2 months, followed by a **24-month linear release**. The releaseEcosystemTokens() function releases these tokens accordingly.

• **Marketing Tokens**: Marketing tokens are distributed using the distributeMarketingTokens() function.

• **Airdrop Tokens**: Airdrop tokens are distributed using the distributeAirdropTokens() function.

7. Buy/Sell Tax System

• **Buy Tax**: 3%

• **Sell Tax**: 4%

• **Burn Rate**: A 1% burn is applied to every swap (buy/sell), with the burned tokens sent to the 0xdead burn address.

• **Tax Exemption**: The contract allows the owner to mark addresses as tax-exempt using the setTaxExempt() function.

• **Fee Wallet**: Taxes are collected and sent to the fee wallet, which can be updated by the owner using updateFeeWallet().

8. Burning Mechanism

• **Burning**: A 1% burn rate is applied on every swap, and the tokens are sent to the burn address (0xdead). The owner can modify the burn rate with setBurnRate().

9. Security Features

• **Pausable**: The contract can pause/unpause trading using pauseTrading() and unpauseTrading() to temporarily disable token transfers.

• **Reentrancy Guard**: Critical functions like presale participation, staking, and liquidity management are protected against reentrancy attacks.

• **Ownable**: Restricted access to sensitive functions ensures only the owner can execute them.

10. Additional Functionality

• **Withdraw ETH**: Collected fees can be withdrawn by the owner using the withdrawETH() function.

• **Stuck Token Withdrawal**: The owner can withdraw any tokens stuck in the contract (except BETC tokens) using the withdrawStuckToken() function.

• **Automated Market Maker Pair Management**: The owner can manage the automated market maker pairs with the setAutomatedMarketMakerPair() function.

Important Functions

For Users

1. participateInPresale(): Allows users to participate in the ongoing presale phase and purchase tokens based on the current phase.
2. stakeTokens(uint256 _amount, uint256 _stakingPeriod): Stake tokens for a specified period to earn rewards.
3. claimStakedRewards(): Claim the staking rewards after the staking period has ended.
4. transfer(address to, uint256 amount): Transfer tokens to another address.
5. approve(address spender, uint256 amount): Approve another address to spend tokens.

For Owner/Admin

1. endPresale(): End the presale phase.
2. createPairAndAddLiquidity(uint256 tokenAmount, uint256 ethAmount, address router): Create a PancakeSwap pair and add liquidity.
3. removeLiquidity(): Remove liquidity after the lock period.
4. distributeTeamAdvisorTokens(address memberOrAdvisor): Distribute vested tokens to team members or advisors based on their vesting schedules.
5. releaseEcosystemTokens(): Release ecosystem tokens according to the vesting schedule.
6. distributeMarketingTokens(address recipient, uint256 amount): Distribute marketing tokens.
7. distributeAirdropTokens(address recipient, uint256 amount): Distribute airdrop tokens.
8. **setBuyTax(uint256 _buyTax) and setSellTax(uint256 _sellTax): Update the buy and sell tax percentages.
9. setTaxExempt(address account, bool exempt): Set the tax-exemption status for an address.
10. pauseTrading() **and** unpauseTrading(): Pause or unpause token transfers.

Security Considerations

1. **OpenZeppelin Libraries**: The contract uses OpenZeppelin's well-audited libraries, which are industry standards for security.
2. **Reentrancy Protection**: Critical functions are protected against reentrancy attacks to ensure no exploit can occur.
3. **Owner Control**: The owner has control over critical aspects of the contract, such as token distribution, taxes, and liquidity. Trust in the owner is vital.
4. **Allocation Limits**: Safeguards are in place to prevent exceeding token allocation limits for staking, vesting, presale, and other distributions.
5. **Liquidity Lock**: Liquidity is locked for a period, offering protection against potential rug pulls.

Conclusion

The Betscoin (BETC) smart contract provides a comprehensive set of features for presale management, staking, liquidity provision, and token

distribution, while maintaining security through OpenZeppelin's libraries and robust mechanisms. The flexibility offered to the contract owner, alongside the security features, ensures both control and protection for users.