



# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



**BabyPolkin**  
\$BabyPolkin

15/01/2024

# TOKEN OVERVIEW

---

## Fees

- Buy fees: 5%
- Sell fees: 5%
- Transfer fees: 0%

At launch, specific conditions apply to buying and selling transactions (check page 8)

## Fees privileges

- Can't change fees

## Ownership

- Owned

## Minting

- No mint function

## Max Tx Amount / Max Wallet Amount

- Can't change max tx amount and / or max wallet amount

## Blacklist

- Blacklist function not detected

## Other privileges

- Can exclude / include from fees
  - Contract owner has to call enableTrading function to enable trade
-

# TABLE OF CONTENTS

1

DISCLAIMER

2

INTRODUCTION

3

WEBSITE + SOCIALS

4-5

AUDIT OVERVIEW

6-9

OWNER PRIVILEGES

10

CONCLUSION AND ANALYSIS

11

TOKEN DETAILS

12

BABYPOLKIN TOKEN ANALYTICS &  
TOP 10 TOKEN HOLDERS

13

TECHNICAL DISCLAIMER



# DISCLAIMER

The information provided on this analysis document is only for general information and should not be used as a reason to invest.

FreshCoins Team will take no payment for manipulating the results of this audit.

The score and the result will stay on this project page information on our website <https://freshcoins.io>

FreshCoins Team does not guarantees that a project will not sell off team supply, or any other scam strategy ( RUG or Honeypot etc )



# INTRODUCTION

**FreshCoins** (Consultant) was contracted by **BabyPolkin** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

**0x003a31a551107d24bded0079790e8c92f195455c**

Network: **Binance Smart Chain (BSC)**

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on **15/01/2024**



# WEBSITE DIAGNOSTIC

<https://babypolkin.com/>



0-49



50-89



90-100



Performance



Accessibility



Best  
Practices



SEO



Progressive  
Web App

## Socials



Twitter

<https://twitter.com/babypolkin>



Telegram

<https://t.me/babypolkin>

# AUDIT OVERVIEW



Security Score



Static Scan

Automatic scanning for common vulnerabilities



ERC Scan

Automatic checks for ERC's conformance



High



Medium



Low



Optimizations



Informational



No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passed
2	Reentrancy and Cross-function	Passed
3	Front running	Passed
4	Timestamp dependence	Passed
5	Integer Overflow and Underflow	Passed
6	Reverted DoS	Passed
7	DoS with block gas limit	Low
8	Methods execution permissions	Passed
9	Exchange rate impact	Passed
10	Malicious Event	Passed
11	Scoping and Declarations	Passed
12	Uninitialized storage pointers	Passed
13	Design Logic	Passed
14	Safe Zeppelin module	Passed



# OWNER PRIVILEGES

- Contract owner can't mint tokens after initial contract deploy
- Contract owner can't exclude an address from transactions
- Contract owner can exclude/include wallet from tax

```
function excludeFromFees(address account, bool excluded) external onlyOwner{
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}
```

- Contract owner can change feeReceiver address

Current value:

feeReceiver: 0xEac0813f7D1DB667DB7dA9E0c365819E3A38166a

```
function changeFeeReceiver(address _feeReceiver) external onlyOwner{
    require(_feeReceiver != address(0), "CSLT: Fee receiver cannot be the zero address");
    feeReceiver = _feeReceiver;

    emit FeeReceiverChanged(feeReceiver);
}
```

- Contract owner can change swap settings (with threshold)

```
function setSwapTokensAtAmount(uint256 newAmount, bool _swapEnabled) external onlyOwner{
    require(newAmount > totalSupply() / 1_000_000, "CSLT: SwapTokensAtAmount must be greater than 0.0001% of total supply");
    swapTokensAtAmount = newAmount;
    swapEnabled = _swapEnabled;

    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}
```

- Contract owner has ability to retrieve any token held by the contract

Native tokens excluded

```
function claimStuckTokens(address token) external onlyOwner {
    require(token != address(this), "CSLT: Owner cannot claim contract's balance of its own tokens");
    if (token == address(0x0)) {
        payable(msg.sender).sendValue(address(this).balance);
        return;
    }

    IERC20(token).transfer(msg.sender, IERC20(token).balanceOf(address(this)));
}
```

## ● Contract owner has to call `enableTrading` function to enable trade

Please note that any wallet excluded from fees retains the ability to engage in trading, even in situations where trading has been disabled

```
function enableTrading() external onlyOwner{
    require(!tradingEnabled, "CSLT: Trading already enabled.");
    tradingEnabled = true;
    swapEnabled = true;
    tradingTime = block.timestamp;

    emit TradingEnabled(tradingEnabled);
}

_transferFrom function line 383
.
.
.
require(tradingEnabled || _isExcludedFromFees[from] || _isExcludedFromFees[to], "CSLT: Trading not yet
enabled!");
.
.
.
```

## ● Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

## ● Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}
```

## Buy Transactions (from UniswapV2Pair):

If the sender is the Uniswap V2 pair (`from == uniswapV2Pair`), the following conditions apply:

If the current block timestamp is greater than **30 minutes** after the trading started

(`block.timestamp > tradingTime + 30 minutes`), then `_totalFees` is set to `feeOnBuy`.

Otherwise, if the above condition is not met, `_totalFees` is set to 0.

## Sell Transactions (to UniswapV2Pair):

If the recipient is the Uniswap V2 pair (`to == uniswapV2Pair`), the following conditions apply:

If the current block timestamp is greater than **30 minutes** after the trading started

(`block.timestamp > tradingTime + 30 minutes`), then `_totalFees` is set to `feeOnSell`.

If the above condition is not met, and the current block timestamp is greater than **15 minutes** after the trading started (`block.timestamp > tradingTime + 15 minutes`), then `_totalFees` is set to 8.

If neither of the above conditions is met, `_totalFees` is set to 10.

## Other Transactions:

For transactions not involving the Uniswap pair, `_totalFees` is set to `feeOnTransfer`.

## Exclusion from Fees or Swapping:

If the sender (`from`) or the recipient (`to`) is excluded from fees (`_isExcludedFromFees[from]` or `_isExcludedFromFees[to]` is `true`), or if swapping is in progress (`swapping` is `true`), then `_totalFees` is set to 0 (no fees).

## Summary:

- When a user buys (`from == uniswapV2Pair`), they will pay a buying fee (`feeOnBuy`) if at least **30 minutes** have passed since trading started.
- When a user sells (`to == uniswapV2Pair`), the fee depends on the time since trading started:
  - If more than **30 minutes** have passed, the **selling fee is `feeOnSell`**.
  - If between **15 and 30 minutes** have passed, the **selling fee is 8**.
  - If less than **15 minutes** have passed, the **selling fee is 10**.
- For other transactions, there is a general transfer fee (`feeOnTransfer`).

```
if (_isExcludedFromFees[from] || _isExcludedFromFees[to] || swapping) {
    _totalFees = 0;
} else if (from == uniswapV2Pair) {
    if (block.timestamp > tradingTime + 30 minutes){
        _totalFees = feeOnBuy;
    } else {
        _totalFees = 0;
    }
} else if (to == uniswapV2Pair) {
    if (block.timestamp > tradingTime + 30 minutes){
        _totalFees = feeOnSell;
    } else if (block.timestamp > tradingTime + 15 minutes){
        _totalFees = 8;
    } else {
        _totalFees = 10;
    }
} else {
    _totalFees = feeOnTransfer;
}
```

### **Recommendation:**

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. The risk can be prevented by temporarily locking the contract or renouncing ownership.



# CONCLUSION AND ANALYSIS



Smart Contracts within the scope were manually reviewed and analyzed with static tools.



Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.



Found 1 HIGH issues during the first review.

# TOKEN DETAILS

## Details

Buy fees: 5%

Sell fees: 5%

Transfer fees: 0%

At launch, specific conditions apply to buying and selling transactions (check page 8)

Max TX: N/A

Max Sell: N/A

## Honeypot Risk

Ownership: Owned

Blacklist: Not detected

Modify Max TX: Not detected

Modify Max Sell: Not detected

Disable Trading: Not detected

## Rug Pull Risk

Liquidity: N/A

Holders: 100% unlocked tokens



# BABYPOLKIN TOKEN ANALYTICS

## & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	0x0Aed73...952A7aDD	420,000,000,000,000	100.0000%

# TECHNICAL DISCLAIMER

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. The audit can't guarantee the explicit security of the audited project / smart contract.

