



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Baby CEO 2.0
\$BABYCEO2.0

06/07/2023



TOKEN OVERVIEW

Fees

- Buy fees: 1%
- Sell fees: 1%
- Transfer fees: 1%

Fees privileges

- Can change buy fees up to 10% and sell fees up to 10%

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

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

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

BABYCEO2.0 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 **Baby CEO 2.0** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xfB6F1CDb75F924237Cd5cf52AF8a582336b33eBd

Network: **Ethereum (ETH)**

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on **06/07/2023**



WEBSITE DIAGNOSTIC

<https://www.babyceo.world/>



0-49



50-89



90-100



Performance



Accessibility



Best
Practices



SEO



Progressive
Web App

Socials



Twitter

<https://twitter.com/BabyCEOx1000>



Telegram

<https://t.me/BabyCEOChinesechat>

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	Passed
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 {
    require(!_isExcludedFromFees[account] != excluded, "Account is already the value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

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

- Contract owner can exclude/include wallet from rewards

```
function excludeFromReward(address account) public onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded");
    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already included");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- Contract owner can change swap settings

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner() {
    require(newAmount > totalSupply() / 1e5, "SwapTokensAtAmount must be greater than 0.001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(newAmount);
}

function setSwapEnabled(bool _enabled) external onlyOwner {
    swapEnabled = _enabled;
    emit SwapEnabledUpdated(_enabled);
}
```

● Contract owner can change buy fees up to 10%, sell fees up to 10%

```
function setBuyFeePercentages(uint256 _taxFeeonBuy, uint256 _buybackBurnFeeOnBuy, uint256 _marketingFeeonBuy) external onlyOwner {
    taxFeeonBuy = _taxFeeonBuy;
    buybackBurnFeeOnBuy = _buybackBurnFeeOnBuy;
    marketingFeeonBuy = _marketingFeeonBuy;
    totalBuyFees = _taxFeeonBuy + _buybackBurnFeeOnBuy + _marketingFeeonBuy;
    require(totalBuyFees <= 10, "Buy fees cannot be greater than 10%");
    emit BuyFeesChanged(taxFeeonBuy, buybackBurnFeeOnBuy, marketingFeeonBuy);
}

function setSellFeePercentages(uint256 _taxFeeonSell, uint256 _buybackBurnFeeOnSell, uint256 _marketingFeeonSell) external onlyOwner {
    taxFeeonSell = _taxFeeonSell;
    buybackBurnFeeOnSell = _buybackBurnFeeOnSell;
    marketingFeeonSell = _marketingFeeonSell;
    totalSellFees = _taxFeeonSell + _buybackBurnFeeOnSell + _marketingFeeonSell;
    require(totalSellFees <= 10, "Sell fees cannot be greater than 10%");
    emit SellFeesChanged(taxFeeonSell, buybackBurnFeeOnSell, marketingFeeonSell);
}
```

● Contract owner can change marketingWallet address

Current value:

marketingWallet: 0xC961C8A0FEC577d9b27d2889721211b6BEC54f2C

```
function changeMarketingWallet(address _marketingWallet) external onlyOwner {
    require(_marketingWallet != marketingWallet, "Marketing wallet is already that address");
    require(_marketingWallet != address(0), "Marketing wallet is the zero address");
    marketingWallet = _marketingWallet;
    emit MarketingWalletChanged(marketingWallet);
}
```

● Contract owner has to call enableTrading() function to enable trade

Please note that any wallet excluded from fees/taxes can still engage in trading even if trading is disabled.

```
function enableTrading() external onlyOwner{
    require(tradingEnabled == false, "Trading is already enabled");
    tradingEnabled = true;
}

_transfer function line 696

require(from != address(0), "ERC20: transfer from the zero address");
require(amount > 0, "Transfer amount must be greater than zero");

if(!_isExcludedFromFees[from] && !_isExcludedFromFees[to]) {
    require(tradingEnabled, "Trading is not enabled yet");
}
```

● Contract owner can withdraw tokens from smart contract

Native tokens excluded

```
function claimStuckTokens(address token) external onlyOwner {
    require(token != address(this), "Owner cannot claim native tokens");
    if (token == address(0x0)) {
        payable(msg.sender).sendValue(address(this).balance);
        return;
    }
    IERC20 ERC20token = IERC20(token);
    uint256 balance = ERC20token.balanceOf(address(this));
    ERC20token.transfer(msg.sender, balance);
}
```

● Contract owner can enable / disabled wallet to wallet transfer fee

```
function enableWalletToWalletTransferWithoutFee(bool enable) external onlyOwner {
    require(walletToWalletTransferWithoutFee != enable, "Wallet to wallet transfer without fee is already set to that value");
    walletToWalletTransferWithoutFee = enable;
    emit WalletToWalletTransferWithoutFeeEnabled(enable);
}
```

● 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);
}
```

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:	1%
Sell fees:	1%
Transfer fees:	1%
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:	Clean



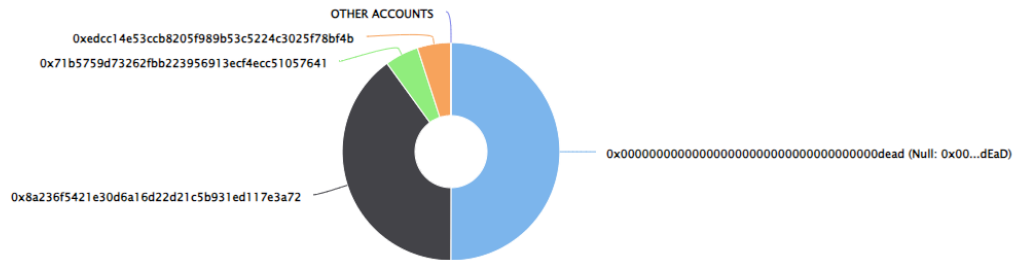
BABYCEO2.0 TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (420,690,000,000,000.00 Tokens) of Baby CEO 2.0

Token Total Supply: 420,690,000,000,000.00 Token | Total Token Holders: 4

Baby CEO 2.0 Top 10 Token Holders

Source: Etherscan.io



(A total of 420,690,000,000,000.00 tokens held by the top 10 accounts from the total supply of 420,690,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Null: 0x00...dEaD	210,469,219,000,000	50.0295%
2	0x8a236f...117E3a72	168,193,850,000,000	39.9805%
3	0x71B575...51057641	21,034,500,000,000	5.0000%
4	0xedCC14...5F78Bf4b	20,992,431,000,000	4.9900%

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.

