



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



BabyGMT
\$BGMT

13/04/2022

TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-6 OWNER PRIVILEGES
- 7 CONCLUSION AND ANALYSIS
- 8 TOKEN DETAILS
- 9 BABYGMT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 10 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 BabyGMT (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x6C5456C5e510D224D436969c1C637c214A01e890

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 13/04/2022



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 dividends

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

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

    emit ExcludeFromDividends(account);
}
```

Contract owner can exclude/include wallet(s) from tax

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(
        _isExcludedFromFees[account] != excluded,
        "TOKEN: Account is already excluded"
    );
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}

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);
}
```

Contract owner can change swap settings

```
function setSwapTokens(uint256 amount) external onlyOwner {
    swapTokensAtAmount = amount;
}
```

Contract owner can change fees up to 100%

```
function setBuyFees(
    uint16 _reward,
    uint16 _marketing,
    uint16 _burn
) external onlyOwner {
    buyFee.reward = _reward;
    buyFee.marketing = _marketing;
    buyFee.burn = _burn;

    totalBuyFee = buyFee.reward + buyFee.marketing;
}

function setSellFees(
    uint16 _reward,
    uint16 _marketing,
    uint16 _burn
) external onlyOwner {
    sellFee.reward = _reward;
    sellFee.marketing = _marketing;
    sellFee.burn = _burn;

    totalSellFee = sellFee.reward + sellFee.marketing;
}
```

Contract owner can change marketingWallet address

Current value:

marketingWallet : 0xb9c25a9e89a15fb380a6fb6eb2d629766f4f8f46

```
function setMarketingWallet(address payable _marketing) external onlyOwner {
    marketingWallet = _marketing;
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _setOwner(address(0));
}
```

Contract owner can transfer ownership

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



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 no issue during the first review.

TOKEN DETAILS

Details

Buy fees: 14%

Sell fees: 17%

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

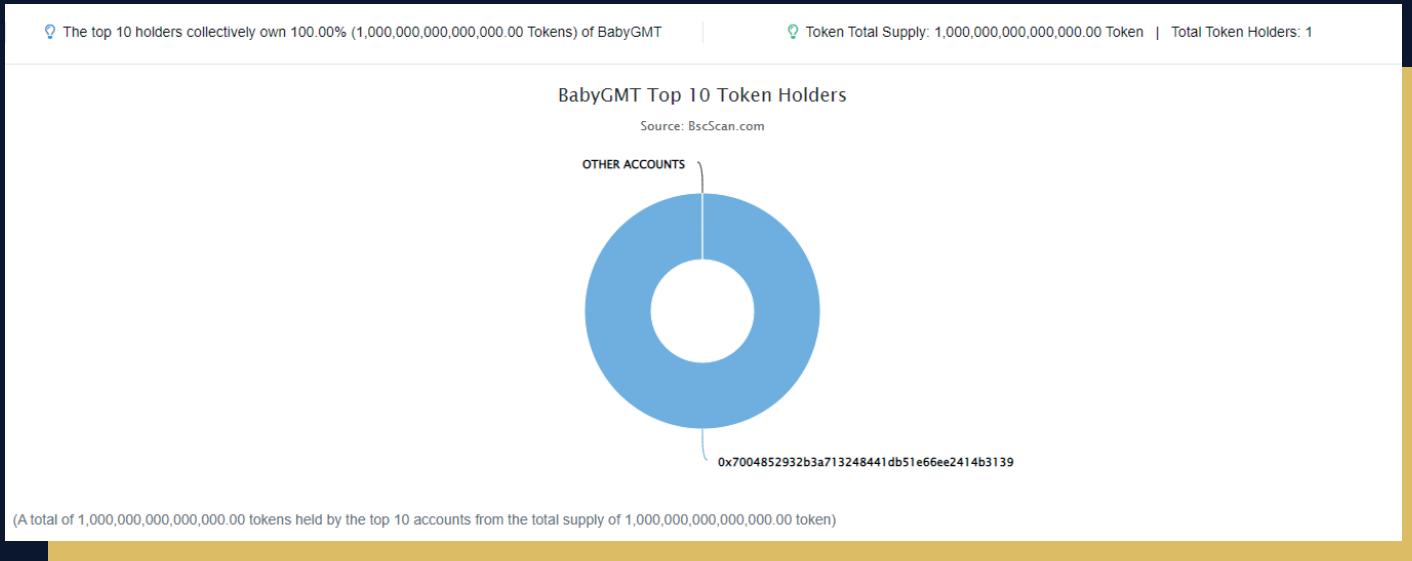
Rug Pull Risk

Liquidity: N/A

Holders: Clean



BGMT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	0x7004852932b3a713248441db51e66ee2414b3139	1,000,000,000,000,000	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.

