



## SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



**Bitcoin Ape**  
\$BAPE



**06/01/2022**



# TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 WEBSITE DIAGNOSTIC
- 5-6 AUDIT OVERVIEW
- 7 OWNER PRIVILEGES
- 8 CONCLUSION AND ANALYSIS
- 9 TOKEN DETAILS
- 10 BITCOIN APE TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS
- 11 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 Bitcoin Ape (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x56C6a8Fd2B3f7d6603d4C28f5049C5e85219b530

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 06/01/2022



# WEBSITE DIAGNOSTIC

<https://bape.digital/>



0-49



50-89



90-100



Performance



Accessability



Best Practices



SEO



Progressive  
Web App

## Metrics



First Contentful Paint

**3.1 s**



Time to interactive

**101.8 s**



Speed Index

**12.2 s**



Total Blocking Time

**102,140 ms**



Large Contentful Paint

**12.1 s**



Cumulative Layout Shift

**0**

# Issues found

---

Serve images in next-gen formats

---

Properly size images

---

Eliminate render-blocking resources

---

Reduce initial server response time

---

Reduce JavaScript execution time - 20,9 s

---

Ensure text remains visible during webfont load

---

Background and foreground colors do not have a sufficient contrast ratio

---

# AUDIT OVERVIEW



**Security Score**



**Static Scan**  
Automatic scanning for common vulnerabilities



**ERC Scan**  
Automatic checks for ERC's conformance

0 **High**

0 **Medium**

0 **Low**

0 **Optimizations**

0 **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 exclude/include from fees a wallet or multiple wallets**

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(
        _isExcludedFromFees[account] != excluded,
        "BABYTOKEN: Account is already the value of '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 tx limits**

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

**Contract owner can change taxes**

```
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%");
}
```

# 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: 10%

Sell fees: 10%

Max TX: N/A

Max Sell: N/A

## Honeypot Risk

Ownership: Owned

Blacklist: Not detected

Modify Max TX: Detected

Modify Max Sell: Not detected

Disable Trading: Not detected

## Rug Pull Risk

Liquidity: N/A

Holders: Clean



# BITCOIN APE TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

💡 The top 100 holders collectively own 100.00% (1,000,000,000,000.00 Tokens) of Bitcoin Ape

💡 Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 1

Bitcoin Ape Top 100 Token Holders

Source: BscScan.com



(A total of 1,000,000,000,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x3ff4a586513e2eb640a1bde6f5e604583069e9fd	1,000,000,000,000	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.

