



# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



**Kong Ape**  
\$KONGAPE

**26/09/2023**



# TOKEN OVERVIEW

---

## Fees

- Buy fees: 8%
- Sell fees: 8%

## Fees privileges

- Can change buy fees up to 7% and sell fees up to 7%

## Ownership

- Owned

## Minting

- No mint function

## Max Tx Amount / Max Wallet Amount

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

## Blacklist

- Blacklist function not detected

## Other privileges

- Can exclude / include from fees
-

# TABLE OF CONTENTS

1

DISCLAIMER

2

INTRODUCTION

3

WEBSITE + SOCIALS

4-5

AUDIT OVERVIEW

6-8

OWNER PRIVILEGES

9

CONCLUSION AND ANALYSIS

10

TOKEN DETAILS

11

KONGAPE TOKEN ANALYTICS &  
TOP 10 TOKEN HOLDERS

12

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 **Kong Ape** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

**0x57800C6Fb76761C4D56e028982e92d23d04800aD**

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 **26/09/2023**



# WEBSITE DIAGNOSTIC

<https://kongape.xyz/>



0-49



50-89



90-100



Performance



Accessibility



Best  
Practices



SEO



Progressive  
Web App

## Socials



Twitter

<https://twitter.com/kongapebsc>



Telegram

<https://t.me/kongapebsc>

# 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	Low
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(s) from tax

```
function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
    for(uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFee[accounts[i]] = excluded;
    }
}
```

- Contract owner can change swap settings

```
function toggleSwap(bool _swapEnabled) public onlyDev {
    swapEnabled = _swapEnabled;
}
```

- Contract owner can change `_developmentAddress` and `_marketingAddress` addresses

Default value:

`_developmentAddress` : `0xaf0cfB234599F4bA5FdDa9129245DF27be0795D5`

`_marketingAddress` : `0x522D51Ace971Ff1749bF942c53D24CcA6A7d8314`

```
event devAddressUpdated(address indexed previous, address indexed adr);
function setNewDevAddress(address payable dev) public onlyDev() {
    emit devAddressUpdated(_developmentAddress, dev);
    _developmentAddress = dev;
    _isExcludedFromFee[_developmentAddress] = true;
}

event marketingAddressUpdated(address indexed previous, address indexed adr);
function setNewMarketingAddress(address payable markt) public onlyDev() {
    emit marketingAddressUpdated(_marketingAddress, markt);
    _marketingAddress = markt;
    _isExcludedFromFee[_marketingAddress] = true;
}
```

- Missing Zero Address Check

- `kongape::setNewDevAddress()`
- `kongape::setNewMarketingAddress()`

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

```
event tokensRescued(address indexed token, address indexed to, uint amount);
function rescueForeignTokens(address _tokenAddr, address _to, uint _amount) public onlyDev() {
    emit tokensRescued(_tokenAddr, _to, _amount);
    Token(_tokenAddr).transfer(_to, _amount);
}
```

## ● Contract owner can set buy fee up to 7% and sell fee up to 7%

```
function setFee(uint256 redisFeeOnBuy, uint256 redisFeeOnSell, uint256 taxFeeOnBuy, uint256 taxFeeOnSell)
public onlyDev {
    require(redisFeeOnBuy < 1, "Redis cannot be more than 2.");
    require(redisFeeOnSell < 1, "Redis cannot be more than 2.");
    require(taxFeeOnBuy < 7, "Tax cannot be more than 8.");
    require(taxFeeOnSell < 7, "Tax cannot be more than 8.");
    _redisFeeOnBuy = redisFeeOnBuy;
    _redisFeeOnSell = redisFeeOnSell;
    _taxFeeOnBuy = taxFeeOnBuy;
    _taxFeeOnSell = taxFeeOnSell;
}
```

## ● Contract owner can transfer ownership

```
event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);
function transferOwnership(address newOwner) public virtual onlyOwner {
    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 no HIGH issues during the first review.

# TOKEN DETAILS

## Details

Buy fees: 8%

Sell fees: 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: Clean



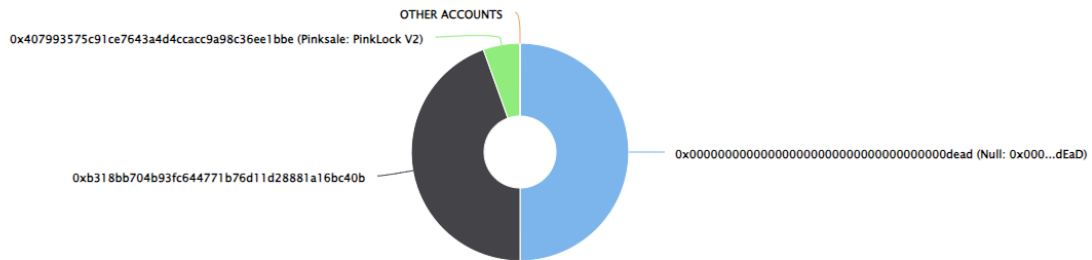
# KONGAPE TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000,000,000.00 Tokens) of Kong Ape

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 3

Kong Ape Top 10 Token Holders

Source: BscScan.com



(A total of 1,000,000,000,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	<a href="#">Null: 0x000...dEaD</a>	500,000,000,000	50.0000%
2	<a href="#">0xB318bB...a16Bc40B</a>	445,350,000,000	44.5350%
3	<a href="#">Pinksale: PinkLock V2</a>	54,650,000,000	5.4650%

# 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.

