



# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



## PEPE BSC 2.0

\$Pepe 2.0

08/07/2023

# TOKEN OVERVIEW

---

## Fees

- Buy fees: 1%
- Sell 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 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

6-9

OWNER PRIVILEGES

10

CONCLUSION AND ANALYSIS

11

TOKEN DETAILS

12

PEPE 2.0 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 **PEPE BSC 2.0** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

**0xc17267Ba4952E2d11547849805DDA4fE0fa100A3**

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 **08/07/2023**



# WEBSITE DIAGNOSTIC

<https://pepebsc20.vip/>



0-49



50-89



90-100



Performance



Accessibility



Best  
Practices



SEO



Progressive  
Web App

## Socials



Twitter

[https://twitter.com/pepebsc\\_20](https://twitter.com/pepebsc_20)



Telegram

<https://t.me/pepebsc20>

# 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 addWhitelisted(address userAddress) external onlyOwner {
    _whitelist[userAddress] = true;
    emit AddWhitelist(userAddress);
}

function removeWhitelisted(address userAddress) external onlyOwner {
    _whitelist[userAddress] = false;
    emit RemoveWhitelist(userAddress);
}
```

- **Contract owner can change swap settings**

```
function setSwapPath(address[] calldata swapPath) external onlyOwner {
    _swapPath = swapPath;
}

function setSwapAddress(address swapRouter) external onlyOwner {
    _swapRouter = swapRouter;
}
```

- **Contract owner can change taxAddress address**

Current value:

**taxAddress** : 0xac0deecfc34e9709b45aa3dcbcff2b2bb55c033b

```
function setTaxAccount(address taxAccount) external onlyOwner {
    require(taxAccount != address(0), "taxAccount_ can't be the zero address");
    _taxAddress = taxAccount;
    _whitelist[taxAccount] = true;
    emit TaxAddressUpdated(taxAccount);
    emit AddWhitelist(taxAccount);
}
```

- **Contract owner can change buy fees up to 10% and sell fees up to 10%**

**\_maxTax** value 10

 **Constructor Arguments** (ABI-Encoded and is the last bytes of the Contract Creation Code above)

[illegible]

```
function setBuyTax(uint256 currentTaxAmount) external onlyOwner {
    require(currentTaxAmount <= _maxTax, "Tax amount exceeds maxTax");
    _buyTax = currentTaxAmount;
    emit BuyTaxUpdated(currentTaxAmount);
}

function setSellTax(uint256 currentTaxAmount) external onlyOwner {
    require(currentTaxAmount <= _maxTax, "Tax amount exceeds maxTax");
    _sellTax = currentTaxAmount;
    emit SellTaxUpdated(currentTaxAmount);
}
```

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

## Native tokens included

```
function retrieveAll(address token) external onlyOwner {
    IERC20 retrieveToken = IERC20(token);
    uint256 retrieveBalance = retrieveToken.balanceOf(address(this));
    retrieveToken.safeTransfer(_msgSender(), retrieveBalance);
}
```

- **Contract owner can manage pool addresses**

```
function addPool(address poolAddress) external onlyOwner {
    _isPool[poolAddress] = true;
    emit AddPoolAddress(poolAddress);
}

function removePool(address poolAddress) external onlyOwner {
    _isPool[poolAddress] = false;
    emit RemovePoolAddress(poolAddress);
}
```

- **Contract owner can renounce ownership**

```
function renounceOwnership() public virtual onlyOwner {
    _transferOwnership(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");  
    _transferOwnership(newOwner);  
}  
  
function _transferOwnership(address newOwner) internal virtual {  
    address oldOwner = _owner;  
    _owner = newOwner;  
    emit OwnershipTransferred(oldOwner, newOwner);  
}
```

### **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:	1%
Sell 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



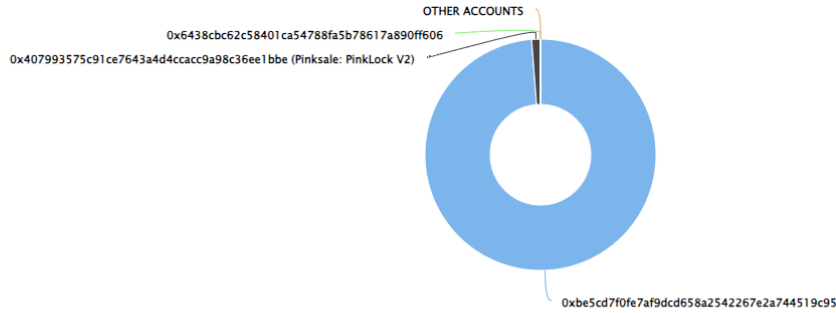
# PEPE 2.0 TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

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

Token Total Supply: 420,690,000,000,000.00 Token | Total Token Holders: 3

PEPE BSC 2.0 Top 10 Token Holders

Source: BscScan.com



(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	0xbe5cd7f0fe7af9dcd658a2542267e2a744519c95	415,587,030,300,000	98.7870%
2	Pinksale: PinkLock V2	5,000,000,000,000	1.1885%
3	0x6438cbc62c58401ca54788fa5b78617a890ff606	102,969,700,000	0.0245%

# 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.

