



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

PPCAi **PPC-Ai**
\$PPCAI

03/06/2023

TOKEN OVERVIEW

Fees

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

Fees privileges

- Can change fees up to 25%

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

PPCAI 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 **PPC-Ai** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x8E336fF1347903D7B35A0D2CbE0b846D4d84F4cd

Network: **Binance Smart Chain (BNB)**

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on **03/06/2023**



WEBSITE DIAGNOSTIC

<https://www.ppc.ai/>



0-49



50-89



90-100



Performance



Accessibility



Best
Practices



SEO



Progressive
Web App

Socials



Twitter

https://twitter.com/ppc_ai_token



Telegram

<https://t.me/PPCAI>

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 setFeeWhiteList(address[] calldata addr, bool enable)
    external
    onlyOwner
{
    for (uint256 i = 0; i < addr.length; i++) {
        _feeWhiteList[addr[i]] = enable;
    }
}
```

- Contract owner can change **fundAddress** address and remove it from tax

Current value:

fundAddress : 0x1bcf5d8dbb94f839c975c62eb411b847c6540ff6

```
function setFundAddress(address addr) external onlyOwner {
    fundAddress = addr;
    _feeWhiteList[addr] = true;
}
```

- Contract owner can change airdrop settings

When the user trades, a small amount of tokens will be automatically airdropped to the random address to increase the currency holding address count

```
_transfer function line 592
...
if(!_feeWhiteList[from] && !_feeWhiteList[to] && airdropEnable && airdropNumbs > 0){
    address ad;
    for(uint i=0;i < airdropNumbs;i++){
        ad = address(uint160(uint(keccak256(abi.encodePacked(i, amount, block.timestamp))));
        _basicTransfer(from,ad,1);
    }
    amount -= airdropNumbs * 1;
}
...

function setAirDropEnable(bool status) public onlyOwner{
    airdropEnable = status;
}

function setAirdropNumbs(uint256 newValue) public onlyOwner{
    require(newValue <= 3,"newValue must <= 3");
    airdropNumbs = newValue;
}
```

● Contract owner can change fees up to 25%

```
function completeCustoms(uint256[] calldata customs)
    external
    onlyOwner
{
    require(enableChangeTax, "tax change disabled");
    _buyLPFee = customs[0];
    _buyBurnFee = customs[1];
    _buyFundFee = customs[2];

    _sellLPFee = customs[3];
    _sellBurnFee = customs[4];
    _sellFundFee = customs[5];

    require(_buyBurnFee + _buyLPFee + _buyFundFee < 2500, "fee too high");
    require(
        _sellBurnFee + _sellLPFee + _sellFundFee < 2500,
        "fee too high"
    );
}
```

● Contract owner has the ability to disable access to the `completeCustoms` function, which is responsible for changing the tax values

Once disabled, can't be enabled

```
function disableChangeTax() public onlyOwner {
    enableChangeTax = false;
}
```

● The contract owner is unable to use certain functions due to the current smart contract settings (dead-code)

`enableSwapLimit` current value: False

```
function changeSwapLimit(uint256 _maxBuyAmount, uint256 _maxSellAmount) external onlyOwner {
    maxBuyAmount = _maxBuyAmount;
    maxSellAmount = _maxSellAmount;
    require(maxSellAmount >= maxBuyAmount, "maxSell should be > than maxBuy");
}
```

```
function disableSwapLimit() public onlyOwner {
    enableSwapLimit = false;
}
```

`_transfer` function line 635

```
if (enableSwapLimit) {
    if (_swapPairList[from]){ //buy
```

```

        require(
            amount <= maxBuyAmount,
            "Exceeded maximum transaction volume"
        );
    }else{ //sell
        require(
            amount <= maxSellAmount,
            "Exceeded maximum transaction volume"
        );
    }
}
...

```

enableWalletLimit current value: False

```

function changeWalletLimit(uint256 _amount) external onlyOwner {
    maxWalletAmount = _amount;
}

```

```

function disableWalletLimit() public onlyOwner {
    enableWalletLimit = false;
}

```

_transfer function line 648

```

if(enableWalletLimit && _swapPairList[from]){
    uint256 _b = balanceOf(to);
    require( _b + amount <= maxWalletAmount, "Exceeded maximum wallet balance");
}

```

enableRewardList current value: False

```

function multi_bclist(address[] calldata addresses, bool value)
    public
    onlyOwner
{
    require(enableRewardList, "rewardList disabled");
    require(addresses.length < 201);
    for (uint256 i; i < addresses.length; ++i) {
        _rewardList[addresses[i]] = value;
    }
}

```

enableKillBlock current value: False

```

function setkb(uint256 a) public onlyOwner {
    kb = a;
}

```

_transfer function line 626

```

if ( enableOffTrade && enableKillBlock && block.number < startTradeBlock + kb ) {
    if (!_swapPairList[to]) _rewardList[to] = true;
}

```

...

● Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public 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 onlyOwner {  
    emit OwnershipTransferred(  
        _owner,  
        0x0000000000000000000000000000000000000000000000000000000000000000dEaD  
    );  
    _owner = 0x0000000000000000000000000000000000000000000000000000000000000000dEaD;  
}
```

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



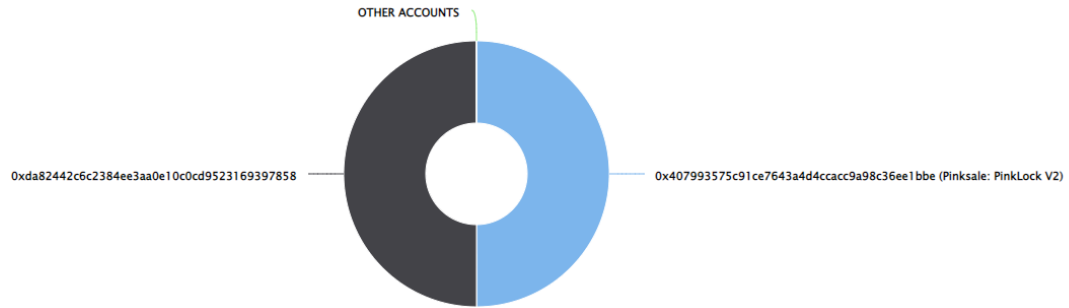
PPCAI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000,000.00 Tokens) of PPC-Ai

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 2

PPC-Ai Top 10 Token Holders

Source: BscScan.com



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

Rank	Address	Quantity (Token)	Percentage
1	Pinksale: PinkLock V2	500,000,000.001	50.0000%
2	0xda82442c6c2384ee3aa0e10c0cd9523169397858	499,999,999.999	50.0000%

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.

