freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

PartnerAI

$PARTAI

06/04/2024

TOKEN OVERVIEW

Fees

• Buy fees:	0%
• Sell fees:	0%

Fees privileges

- Can change buy fees up to 5% and sell fees up to 5%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change max tx amount and max wallet amount (with threshold)

Blacklist

- Blacklist function not detected

Other privileges

- Can exclude / include from fees
- Investors won't be able to sell their tokens on PancakeSwap if swapEnabled is true and _swapTokensAtAmount is set to 0
- Contract owner has to set setTrading function to it’s true value

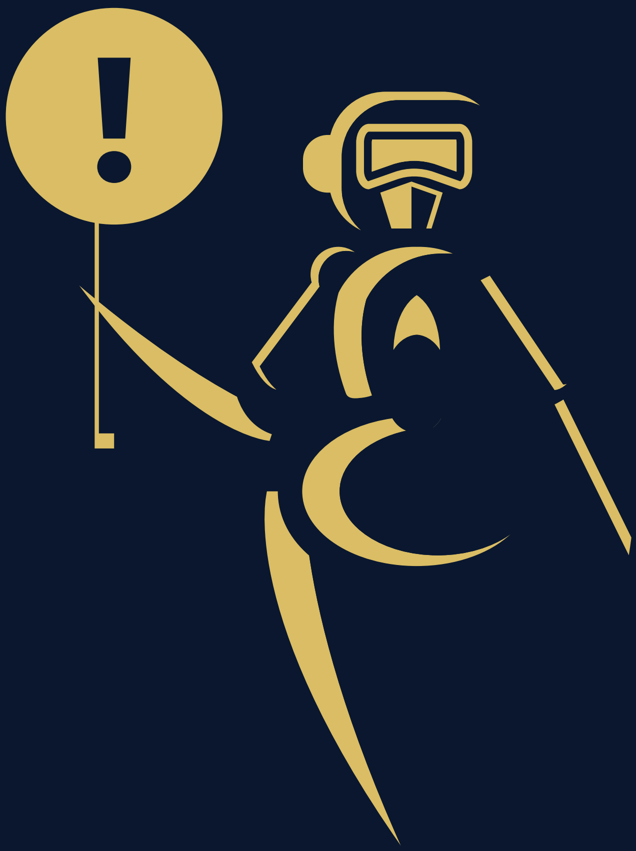
TABLE OF CONTENTS

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 PartnerAI (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xcC2906eFC7bb619895810ee3abf844e73959a60e

Network: Ethereum (ETH)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 06/04/2024

WEBSITE DIAGNOSTIC

https://partnerai.tech/

🟥 0-49 🟨 50-89 🟩 90-100

95	94	91	90	NA
Performance	Accessibility	Best Practices	SEO	Progressive Web App

Socials

Twitter

https://twitter.com/partneraiteam

Telegram

https://t.me/partneraiportal

AUDIT OVERVIEW

66

Security Score

HIGH RISK

Audit FAIL

93

Static Scan

Automatic scanning for common vulnerabilities

62

ERC Scan

Automatic checks for ERC's conformance

2	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't mint tokens after initial contract deploy

🟢 Contract owner can't exclude addresses from transactions

🔴 Contract owner can change swap settings (without threshold)

Users won't be able to sell their tokens on PancakeSwap if swapEnabled is true and _swapTokensAtAmount is set to 0 because every transaction involving the token will trigger a swap for BNB, regardless of the transaction size

Default value: uint256 public _swapTokensAtAmount = 100 * 10**18;

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

function setMinSwapTokensThreshold(uint256 swapTokensAtAmount) public onlyOwner {
        _swapTokensAtAmount = swapTokensAtAmount;
}
```

🔴 Contract owner has to set setTrading function to it’s true value

Please note that owner wallet retains the ability to engage in trading, even in situations where trading has been disabled

```
function setTrading(bool _tradingOpen) public onlyOwner {
      require(!tradingOpen, "Cannot reenable trading");
      tradingOpen = _tradingOpen;
}

_transferFrom function line 338
.
.
.
if (!tradingOpen) {
            require(from == owner(), "TOKEN: This account cannot send tokens until trading is enabled");
}
.
.
.
```

🟢 Contract owner can change max tx amount limitation (with threshold)

```
function setMaxTxnAmount(uint256 maxTxAmount) public onlyOwner {
      require(maxTxAmount >= totalSupply().div(100), "value too low");
       _maxTxAmount = maxTxAmount;
}
```

● Contract owner can change max wallet amount limitation (with threshold)

```
function setMaxWalletSize(uint256 maxWalletSize) public onlyOwner {
        require(maxWalletSize >= totalSupply().div(100), "value too low");
        _maxWalletSize = maxWalletSize;
}
```

● Contract owner can change buy fees up to 5% and sell fees up to 5%

```
function setFee(uint256 redisFeeOnBuy, uint256 redisFeeOnSell, uint256 taxFeeOnBuy, uint256 taxFeeOnSell)
public onlyOwner {
        require(redisFeeOnBuy >= 0 && redisFeeOnBuy <= 0, "Buy rewards must be between 0% and 0%");
        require(taxFeeOnBuy >= 0 && taxFeeOnBuy <= 5, "Buy tax must be between 0% and 5%");
        require(redisFeeOnSell >= 0 && redisFeeOnSell <= 0, "Sell rewards must be between 0% and 0%");
        require(taxFeeOnSell >= 0 && taxFeeOnSell <= 5, "Sell tax must be between 0% and 5%");

        _redisFeeOnBuy = redisFeeOnBuy;
        _redisFeeOnSell = redisFeeOnSell;
        _taxFeeOnBuy = taxFeeOnBuy;
        _taxFeeOnSell = taxFeeOnSell;
}
```

● 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 transfer ownership

```
function transferOwnership(address newOwner) public virtual 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 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 2 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	0%
Sell fees:	0%
Max TX:	1,000,000
Max Wallet:	1,000,000

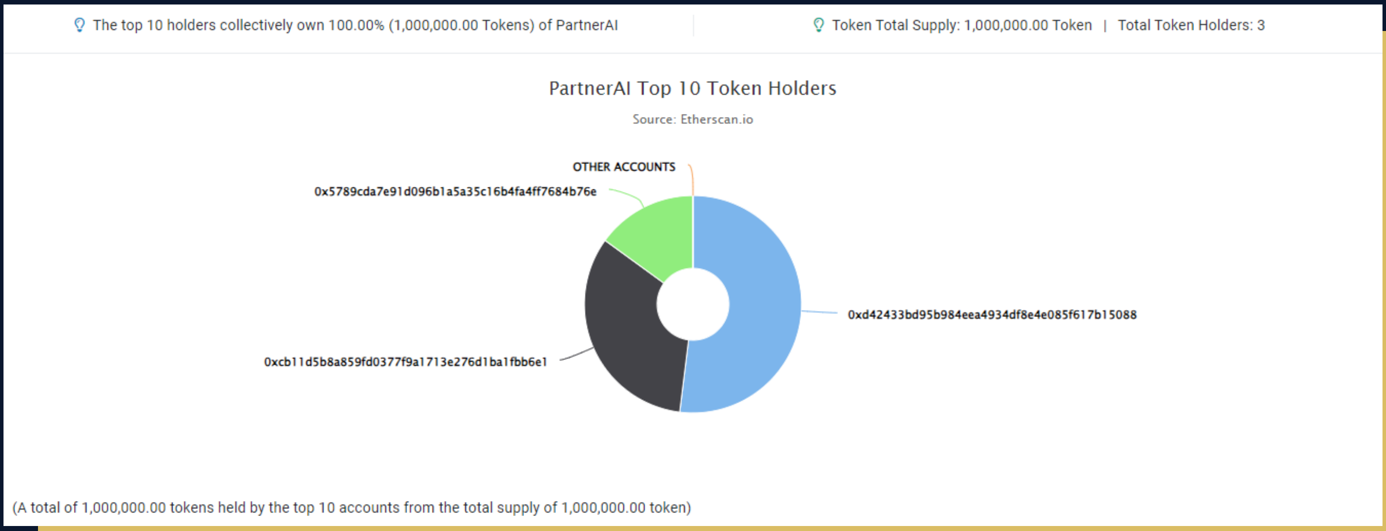
Honeypot Risk

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:	48% unlocked tokens

PARTAI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



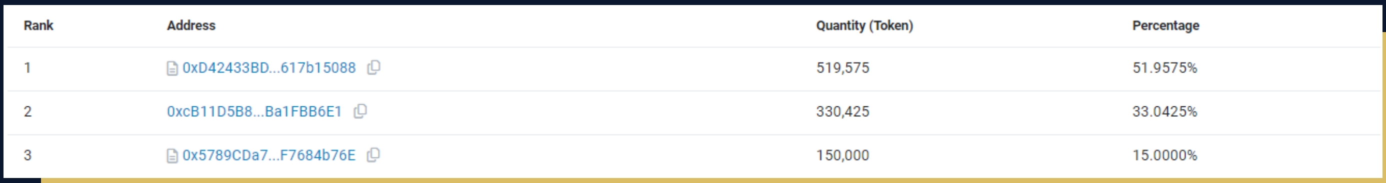
The top 10 holders collectively own 100.00% (1,000,000.00 Tokens) of PartnerAI

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

PartnerAI Top 10 Token Holders

Source: Etherscan.io

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



Rank	Address	Quantity (Token)	Percentage
1	0xD42433BD...617b15088	519,575	51.9575%
2	0xcB11D5B8...Ba1FBB6E1	330,425	33.0425%
3	0x5789CDa7...F7684b76E	150,000	15.0000%

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.