freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

CRYPTO 1996

$CRYPT96

23/11/2022

TOKEN OVERVIEW

Fees

• Buy fees:	7%
• Sell fees:	7%

Fees privileges

- Can change buy fees up to 33% and sell fees up to 100% (page 7)

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change max tx amount without threshold

Blacklist

- Blacklist function detected

Other privileges

- Can exclude / include from fees
- Can exclude / include from dividends
- Can exclude / include from tx limitations and wallet limitations

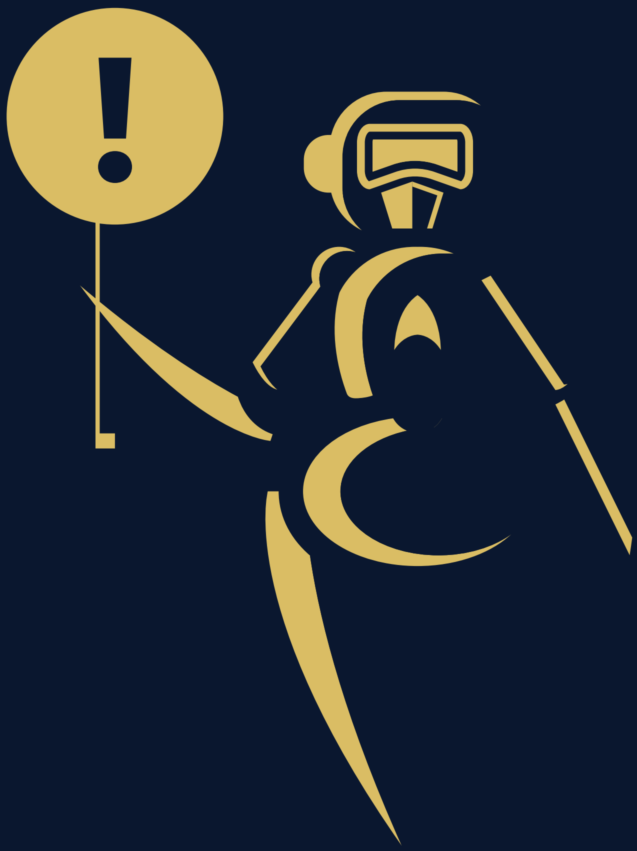
TABLE OF CONTENTS

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
CRYPTO 1996 (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xCc43C923Bbab3b293613cf2Be7251c6c47E9021C

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 23/11/2022

AUDIT OVERVIEW

41

Security Score
HIGH RISK

93

Static Scan
Automatic scanning for common vulnerabilities

30

ERC Scan
Automatic checks for ERC's conformance

5	High
4	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	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 exclude an address from transactions

```
function enable_blacklist(bool _status) public onlyOwner {
        blacklistMode = _status;
}

function manage_blacklist(address[] calldata addresses, bool status) public onlyOwner {
        for (uint256 i; i < addresses.length; ++i) {
            isBlacklisted[addresses[i]] = status;
        }
}
```

🟡 Contract owner can exclude/include wallet from tax

```
function setIsFeeExempt(address holder, bool exempt) external authorized {
        isFeeExempt[holder] = exempt;
}
```

🟡 Contract owner can exclude/include wallet from tx limitations

```
function setIsTxLimitExempt(address holder, bool exempt) external authorized {
        isTxLimitExempt[holder] = exempt;
}
```

🟡 Contract owner can exclude/include wallet from dividends

```
function setIsDividendExempt(address holder, bool exempt) external authorized {
        require(holder != address(this) && holder != pair);
        isDividendExempt[holder] = exempt;
        if(exempt){
            distributor.setShare(holder, 0);
        }else{
            distributor.setShare(holder, _balances[holder]);
        }
}
```

🟡 Contract owner can exclude/include wallet from cooldown between trades limitation

```
function setIsTimelockExempt(address holder, bool exempt) external authorized {
        isTimelockExempt[holder] = exempt;
}
```

🟢 Contract owner can change fees up to 33%

```
function setFees(uint256 _liquidityFee, uint256 _reflectionFee, uint256 _marketingFee, uint256 _devFee,
uint256 _feeDenominator) external authorized {
        liquidityFee = _liquidityFee;
        reflectionFee = _reflectionFee;
        marketingFee = _marketingFee;
        devFee = _devFee;
        totalFee = _liquidityFee.add(_reflectionFee).add(_marketingFee).add(_devFee);
        feeDenominator = _feeDenominator;
        require(totalFee < feeDenominator/3, "Fees cannot be more than 33%");
}
```

🟢 Contract owner can change autoLiquidityReceiver, marketingFeeReceiver and devFeeReceiver addresses

Current values:

autoLiquidityReceiver : 0x3c3e3125bc869d5f8c4223d7c41fbb8c8e2e1146

marketingFeeReceiver : 0xceea66dc4c15f1ac4b3149d0dbfb090bff22f2e9

devFeeReceiver : 0xceea66dc4c15f1ac4b3149d0dbfb090bff22f2e9

```
function setFeeReceivers(address _autoLiquidityReceiver, address _marketingFeeReceiver, address _devFeeReceiver ) external authorized {
        autoLiquidityReceiver = _autoLiquidityReceiver;
        marketingFeeReceiver = _marketingFeeReceiver;
        devFeeReceiver = _devFeeReceiver;
}
```

🟢 Contract owner can change swap settings

```
function setSwapBackSettings(bool _enabled, uint256 _amount) external authorized {
        swapEnabled = _enabled;
        swapThreshold = _amount;
}
```

🔴 Contract owner can change max tx amount limitation to 0

Transfers can be disabled if _maxTxAmount value is set to 0

```
function setTxLimit(uint256 amount) external authorized {
        _maxTxAmount = amount;
}
```

🟢 Contract owner can change max tx percent with threshold

```
function setMaxTxPercent_base1000(uint256 maxTXPercentage_base1000) external onlyOwner() {
        _maxTxAmount = (_totalSupply * maxTXPercentage_base1000 ) / 1000;
}
```

🟢 Contract owner can change max wallet percent with threshold

```
function setMaxWalletPercent_base1000(uint256 maxWallPercent_base1000) external onlyOwner() {
    _maxWalletToken = (_totalSupply * maxWallPercent_base1000 ) / 1000;
}
```

🟢 Contract owner can change cooldown between trades status and interval value

```
function cooldownEnabled(bool _status, uint8 _interval) public onlyOwner {
    buyCooldownEnabled = _status;
    cooldownTimerInterval = _interval;
}
```

🔴 Contract owner can set sell multiplier (without threshold)

```
function set_sell_multiplier(uint256 Multiplier) external onlyOwner{
    sellMultiplier = Multiplier;
}
```

🔴 Contract owner can enable/disable trading

```
function tradingStatus(bool _status) public onlyOwner {
    tradingOpen = _status;
}
```

🔴 Contract owner can transfer tokens from any wallet to specific wallet(s)

```
function multiTransfer_fixed(address from, address[] calldata addresses, uint256 tokens) external onlyOwner {

  require(addresses.length < 801,"GAS Error: max airdrop limit is 800 addresses");

  uint256 SCCC = tokens * addresses.length;

  require(balanceOf(from) >= SCCC, "Not enough tokens in wallet");

  for(uint i=0; i < addresses.length; i++){
    _basicTransfer(from,addresses[i],tokens);
    if(!isDividendExempt[addresses[i]]) {
      try distributor.setShare(addresses[i], _balances[addresses[i]]) {} catch {}
    }
  }

// Dividend tracker
  if(!isDividendExempt[from]) {
    try distributor.setShare(from, _balances[from]) {} catch {}
  }
}
```

Contract owner can transfer ownership

```
function transferOwnership(address payable adr) public onlyOwner {
        owner = adr;
        authorizations[adr] = true;
        emit OwnershipTransferred(adr);
}
```

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 5 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	7%
Sell fees:	7%
Max TX:	20,000
Max Sell:	N/A

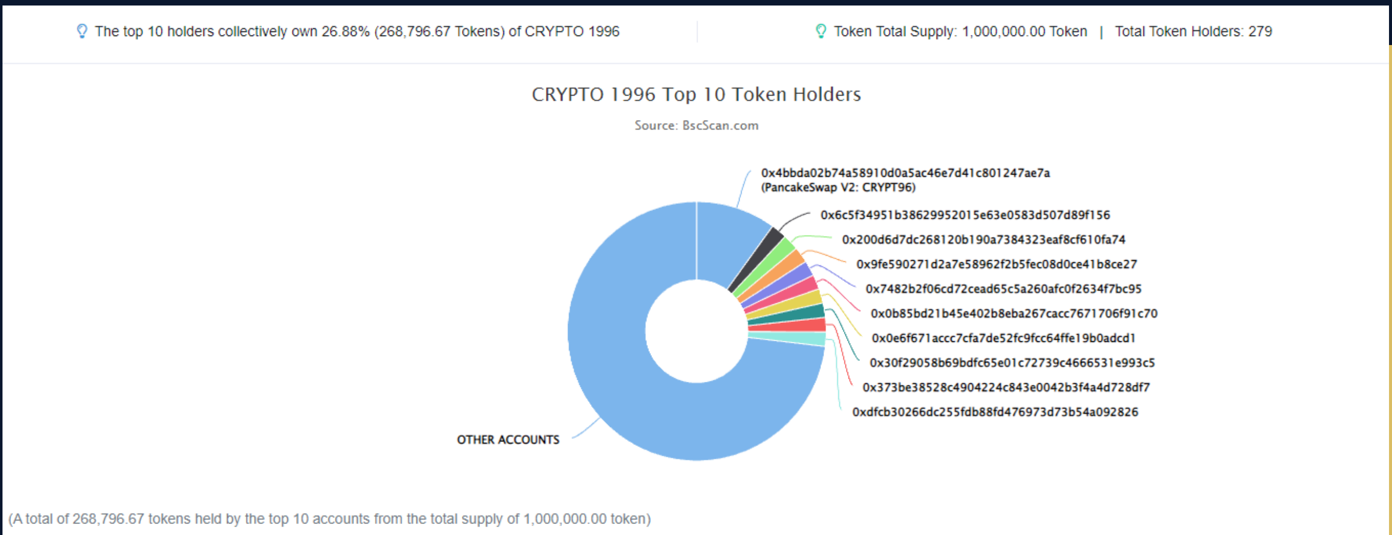
Honeypot Risk

Ownership:	Owned
Blacklist:	Detected
Modify Max TX:	Detected
Modify Max Sell:	Not detected
Disable Trading:	Detected

Others

Liquidity:	N/A
Holders:	Clean

CRYPTO96 TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



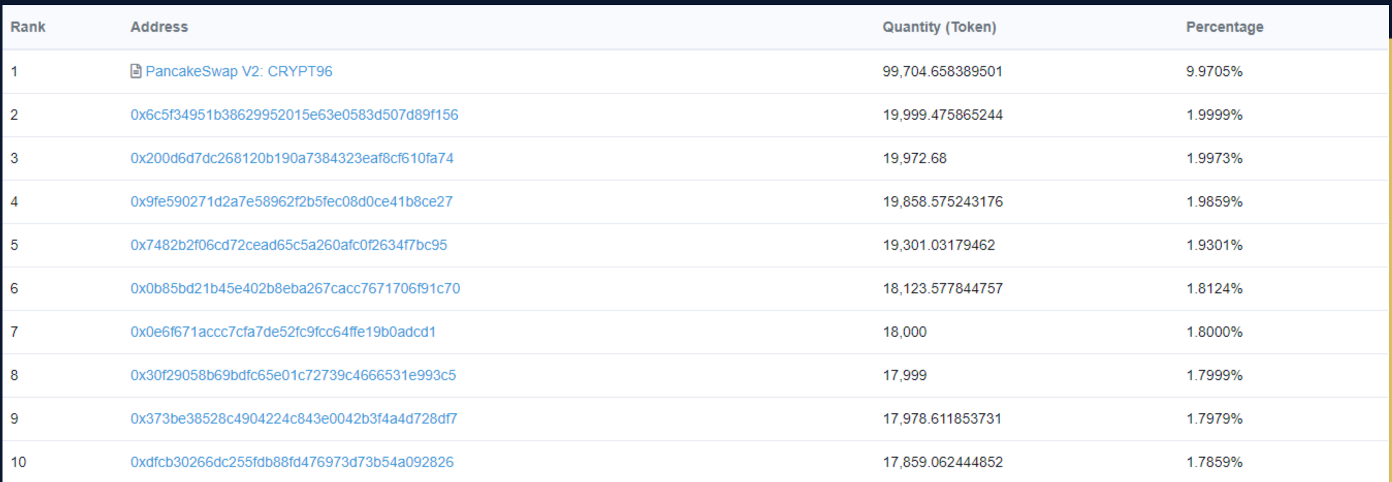
The top 10 holders collectively own 26.88% (268,796.67 Tokens) of CRYPTO 1996

Token Total Supply: 1,000,000.00 Token | Total Token Holders: 279

CRYPTO 1996 Top 10 Token Holders

Source: BscScan.com

(A total of 268,796.67 tokens held by the top 10 accounts from the total supply of 1,000,000.00 token)



Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: CRYPT96	99,704.658389501	9.9705%
2	0x6c5f34951b38629952015e63e0583d507d89f156	19,999.475865244	1.9999%
3	0x200d6d7dc268120b190a7384323eaf8cf610fa74	19,972.68	1.9973%
4	0x9fe590271d2a7e58962f2b5fec08d0ce41b8ce27	19,858.575243176	1.9859%
5	0x7482b2f06cd72cead65c5a260afc0f2634f7bc95	19,301.03179462	1.9301%
6	0x0b85bd21b45e402b8eba267cacc7671706f91c70	18,123.577844757	1.8124%
7	0x0e6f671accc7cfa7de52fc9fcc64ffe19b0adcd1	18,000	1.8000%
8	0x30f29058b69bdfc65e01c72739c4666531e993c5	17,999	1.7999%
9	0x373be38528c4904224c843e0042b3f4a4d728df7	17,978.611853731	1.7979%
10	0xdfcb30266dc255fdb88fd476973d73b54a092826	17,859.062444852	1.7859%

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.