freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

CRYPTO VAULT

Crypto Vault

$CVT

02/03/2022

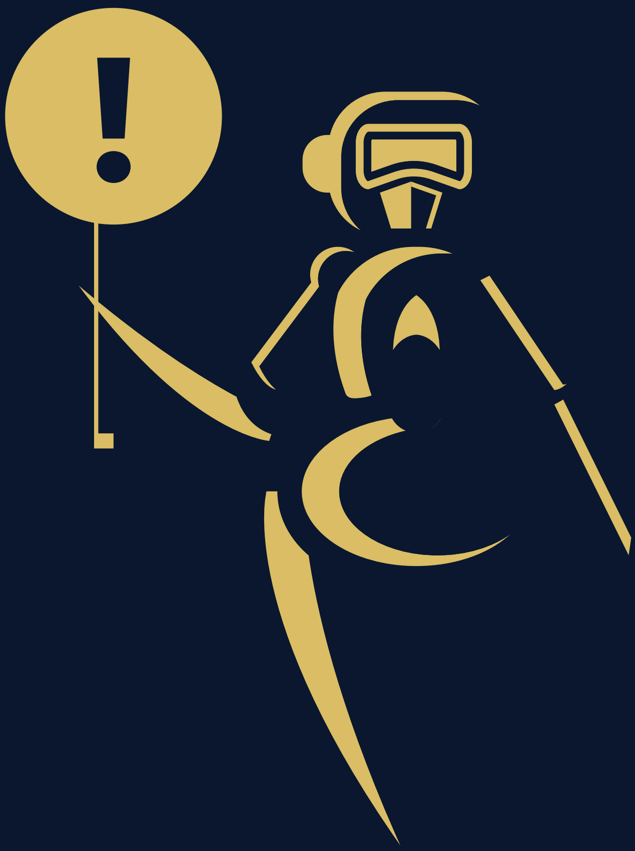
TABLE OF CONTENTS

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

0xF72B0F79dC66f270FE52C67e56e12872F86cae2d

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 02/03/2022

AUDIT OVERVIEW

97

Security Score

95

Static Scan
Automatic scanning for common vulnerabilities

98

ERC Scan
Automatic checks for ERC's conformance

0	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	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 exclude an address from transactions.

Contract owner can't mint tokens after initial contract deploy

Contract owner can exclude/include wallet from tax

```
function excludeFromFee(address account) public onlyOwner {
        _isExcludedFromFee[account] = true;
}

function includeInFee(address account) public onlyOwner {
        _isExcludedFromFee[account] = false;
}
```

Contract owner can exclude/include wallet from reward

```
function excludeFromReward(address account) public onlyOwner {
    require(!_isExcluded[account], 'Account already excluded');
    if (_rOwned[account] > 0) {
      _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

function includeInReward(address account) external onlyOwner {
    require(_isExcluded[account], 'Account is already included');
    for (uint256 i = 0; i < _excluded.length; i++) {
      if (_excluded[i] == account) {
        _excluded[i] = _excluded[_excluded.length - 1];
        _tOwned[account] = 0;
        _isExcluded[account] = false;
        _excluded.pop();
        break;
      }
    }
}
```

Contract owner can exclude/include wallet from tx limitations

```
function setExcludedMaxWallet(address acc, bool value) public onlyOwner {
      _isExcludedFromMaxWalletLimit[acc] = value;
}

function setIsExcludedFromTXLimit(address account, bool isExcluded) public onlyOwner {
    _isExcludedFromTxLimit[account] = isExcluded;
}
```

Contract owner can change _marketingWalletAddress and _buybackWallet addresses

Current values:

_marketingWalletAddress : 0xc5b97f4d56984ac5f5e5fe43f1bdebc02ef00dfc

_buybackWallet : 0x000000000000000000000000000000000000dead

```
function setMarketingAddr(address account) external onlyOwner {
  _marketingWalletAddress = account;
}

function setBuybackWallet(address acc) public onlyOwner {
    _buybackWallet = acc;
}
```

Contract owner can change max tx / wallet tx amount

```
function setMaxWalletAmount(uint val) public onlyOwner {
    require(val > 100000 * 10 **9, "Min wallet reached");
    maxWalletAmount = val;
}

function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner {
    require(maxTxPercent > 0, "min 0 invalid");
  _maxTxAmount = _tTotal.mul(maxTxPercent).div(100 * 10**2);
}
```

Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool toggle) public onlyOwner {
  swapAndLiquifyEnabled = toggle;
  emit SwapAndLiquifyEnabledUpdated(toggle);
}
```

Contract owner can change the fees up to 25%

```
function setSellFee(uint buyback, uint marketing, uint liquidity, uint reflect) public onlyOwner {
    buybackFeeSell = buyback;
    marketingFeeSell = marketing;
    liquidityFeeSell = liquidity;
    reflectFeeSell = reflect;
    require(buyback + marketing + liquidity + reflect <= 25, "max 25%");
}

function setBuyFees(uint buyback, uint marketing, uint liquidity, uint reflect) public onlyOwner {
    buybackFeeBuy = buyback;
    marketingFeeBuy = marketing;
    liquidityFeeBuy = liquidity;
    reflectFeeBuy = reflect;
    require(buyback + marketing + liquidity + reflect <= 25, "max 25%");
}
```

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 issue during the first review.

TOKEN DETAILS

Details

Buy fees:	10%
Sell fees:	10%
Max TX:	10,000,000,000
Max Sell:	N/A

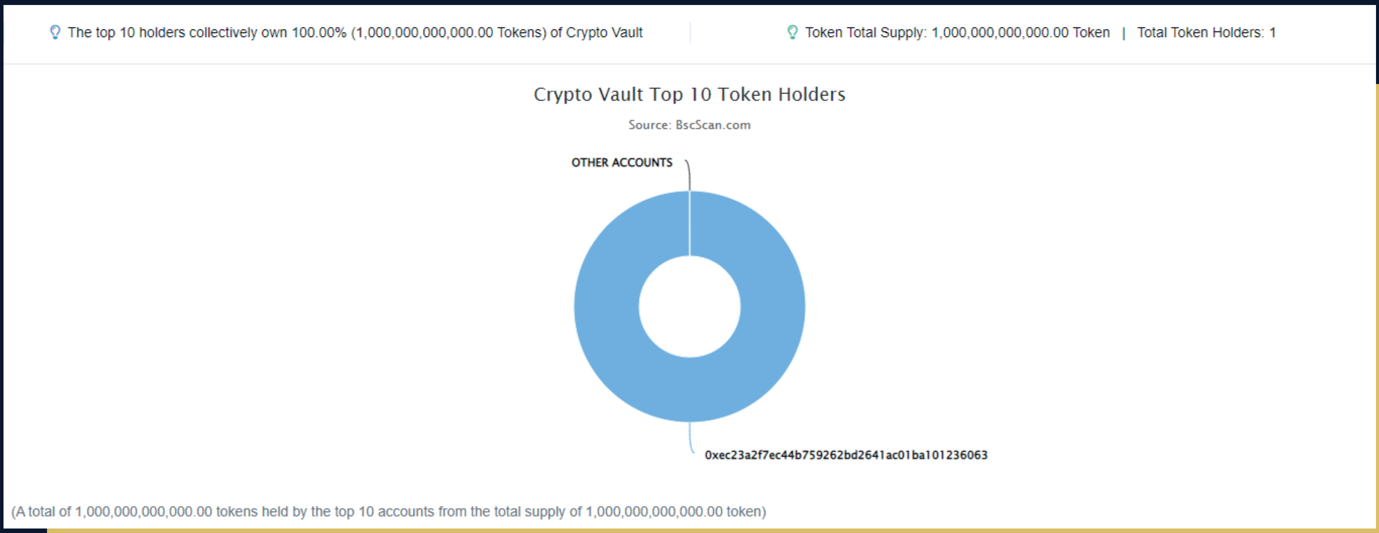
Honeypot Risk

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:	Clean

CRYPTO VAULT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



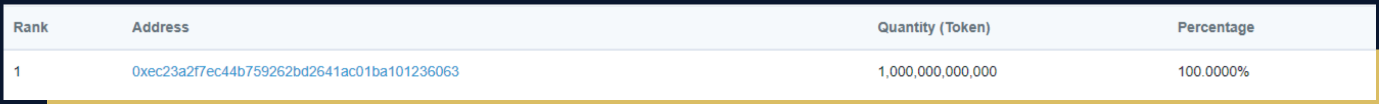
The top 10 holders collectively own 100.00% (1,000,000,000,000.00 Tokens) of Crypto Vault

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

Crypto Vault 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	0xec23a2f7ec44b759262bd2641ac01ba101236063	1,000,000,000,000	100.0000%

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.