



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



AnonVault
\$AnonVault

28/02/2022

TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-6 OWNER PRIVILEGES
- 7 CONCLUSION AND ANALYSIS
- 8 TOKEN DETAILS
- 9 ANONVAULT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 10 TECHNICAL DISCLAIMER



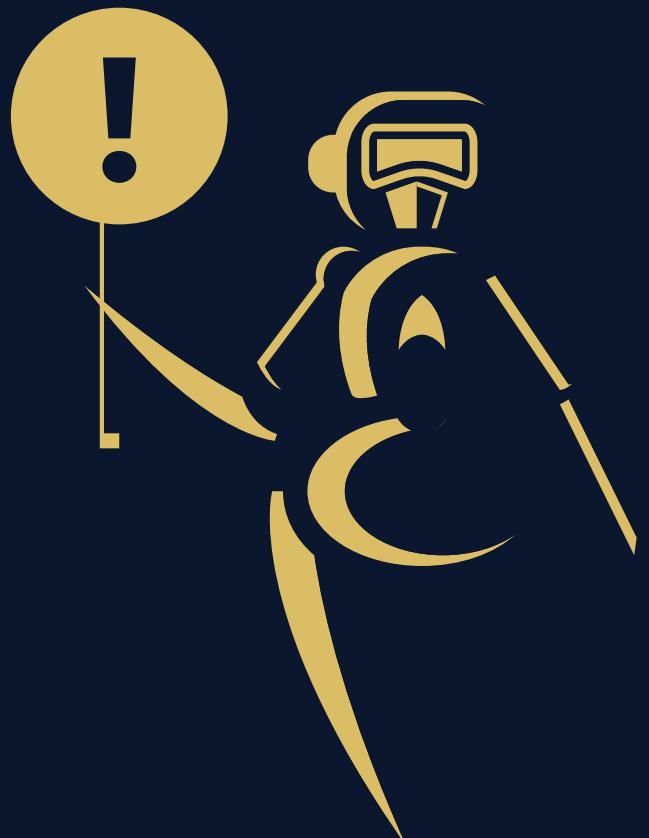
DISCLAIMER

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

0xB8cF7B46506395e8418F84f07c1d521de28cc892

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 28/02/2022



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



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 fees

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

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

Contract owner can change the fees up to 40%

```
/* Fees */
uint256 public _maxLiqFee = 10;
uint256 public _maxTaxFee = 10;

.

.

.

function setTaxFeePercent(uint256 taxFee) external onlyOwner {
    require(taxFee >= 0 && taxFee <=_maxTaxFee,"taxFee out of range");
    _previousTaxFee = _taxFee;
    _taxFee = taxFee;
}

function setLiqFeePercent(uint256 liqFee) external onlyOwner {
    require(liqFee >= 0 && liqFee <=_maxLiqFee,"liquidityFee out of range");
    _previousLiqFee = _liqFee;
    _liqFee = liqFee;
}

function setDeveloperFeePercent(uint256 developerFee) external onlyOwner {
    require(developerFee >= 0 && developerFee <=_maxTaxFee,"taxFee out of range");
    _previousDeveloperFee = _developerFee;
    _developerFee = developerFee;
}

function setMarketingFeePercent(uint256 marketingFee) external onlyOwner {
    require(marketingFee >= 0 && marketingFee <=_maxTaxFee,"taxFee out of range");
    _previousMarketingFee = _marketingFee;
    _marketingFee = marketingFee;
}
```

Contract owner can change `_developerAddress` and `_marketingAddress` addresses

Current values:

`_developerAddress` : 0x00

`_marketingAddress` : 0x00

```
function setDeveloperAddress(address payable developer) public onlyOwner {  
    _developerAddress = developer;  
}
```

```
function setMarketingAddress(address payable marketing) public onlyOwner {  
    _marketingAddress = marketing;  
}
```

Contract owner can change max tx amount

```
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner {  
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(  
        10**2  
    );  
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {  
    emit OwnershipTransferred(_owner, address(0));  
    _owner = 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");  
    emit OwnershipTransferred(_owner, newOwner);  
    _owner = newOwner;  
}
```



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:	5,000,000,000
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



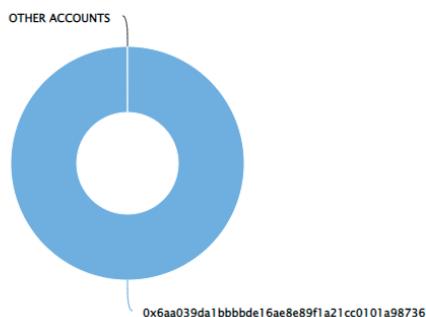
ANONVAULT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

💡 The top 10 holders collectively own 100.00%
(1,000,000,000,000.00 Tokens) of AnonVault (ANONVAULT.COM)

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

AnonVault (ANONVAULT.COM) 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	0x6aa039da1bbbde16ae8e89f1a21cc0101a98736	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.

