freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Shepherd Inu

$SINU

29/07/2022

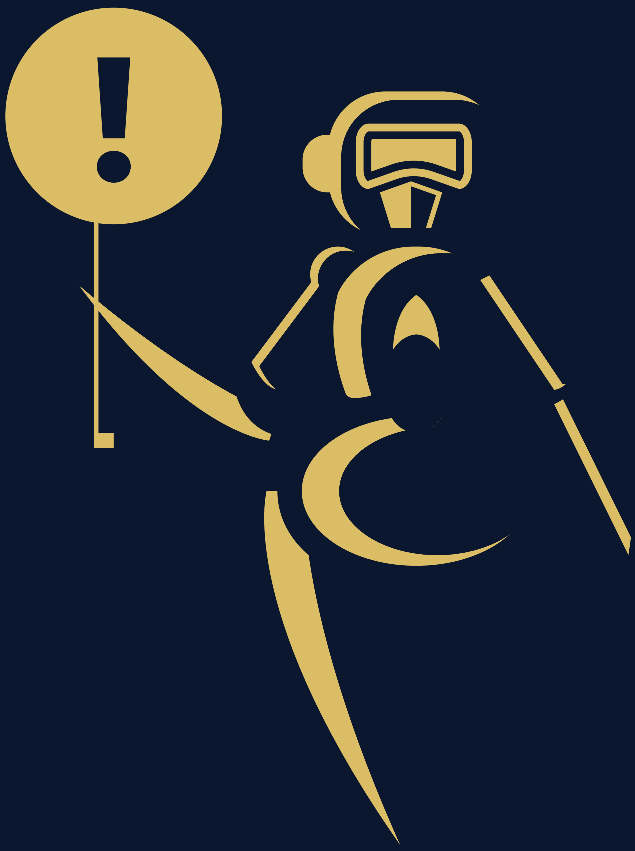
TABLE OF CONTENTS

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

0x91011693694bcb5838ddbc8cb328e6af340b1f13

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 29/07/2022

AUDIT OVERVIEW

82

Security Score

76

Static Scan

Automatic scanning for common vulnerabilities

90

ERC Scan

Automatic checks for ERC's conformance

5	High
6	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 change max wallet amount (with threshold)

```
function setMaxWallet(uint256 amount) external onlyOwner {
      require(amount >= _totalSupply / 1000 );
      _maxWalletSize = amount;
}
```

🟡 Contract owner can change max tx amount

```
function setTxLimit(uint256 amountBuy) external onlyOwner {
      _maxTxAmount = amountBuy;
}
```

🟡 Contract owner can enable/disable cooldown between trades

Current value (interval): 5 uint8

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

Contract owner can enable/disable additional tax for certain wallets

```
.
.
.
uint256 multiplier = isSell ? _sellMultiplier : 100; //dont touch this section
if(taxMode && !istaxed[receiver] && !isSell){
    multiplier = 800;
}
.
.
.
```

Normal multiplier on buy is 1 (value 100), if taxMode is enabled, for wallets included in istaxed array, multiplier will be 8

```
function enable_tax(bool _status) public onlyOwner {
    taxMode = _status;
}

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

Contract owner can set sell multiplier (without threshold)

```
function setSellMultiplier(uint256 multiplier) external onlyOwner{
    _sellMultiplier = multiplier;
}
```

Contract owner can enable/disable trade

```
function tradingstatus(bool state) public onlyOwner {
    tradingOpen = state;
}

function OpenTrading(uint256 _swapAt, uint256 _swapDelay) public onlyOwner {
    tradingOpen = true;
    launchBlock = block.number;
    swapAt = _swapAt * (10 ** 9);
    swapDelay = _swapDelay;
}
```

Contract owner can change swap settings

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

🟢 Contract owner can change marketingFeeReceiver and ecosystemFeeReceiver addresses

Current values:

marketingFeeReceiver : 0xca390d2e5a82e77c9aeef7ec53b25ad4efa5709f

ecosystemFeeReceiver : 0xbaf929601c97efbbe7fe607940fdeadfec05c8be

```
function setFeeReceiver(address _marketingFeeReceiver, address _ecosystemFeeReceiver) external onlyOwner {
        marketingFeeReceiver = _marketingFeeReceiver;
        ecosystemFeeReceiver = _ecosystemFeeReceiver;
}
```

🔴 Contract owner can change fees up to 100%

```
function setFees(uint256 _liquidityFee, uint256 _marketingFee, uint256 _ecosystemFee, uint256 _feeDenominator) external onlyOwner {
        liquidityFee = _liquidityFee;
        marketingFee = _marketingFee;
        ecosystemFee = _ecosystemFee;
        totalFee = _liquidityFee.add(_marketingFee).add(_ecosystemFee);
        feeDenominator = _feeDenominator;
}
```

🟢 Contract owner can renounce ownership

```
function renounceOwnership() public onlyOwner {
        _setOwner(address(0));
}
```

🟢 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:	9%
Sell fees:	9% (more details on page 6)
Max TX:	20,000
Max Sell:	N/A

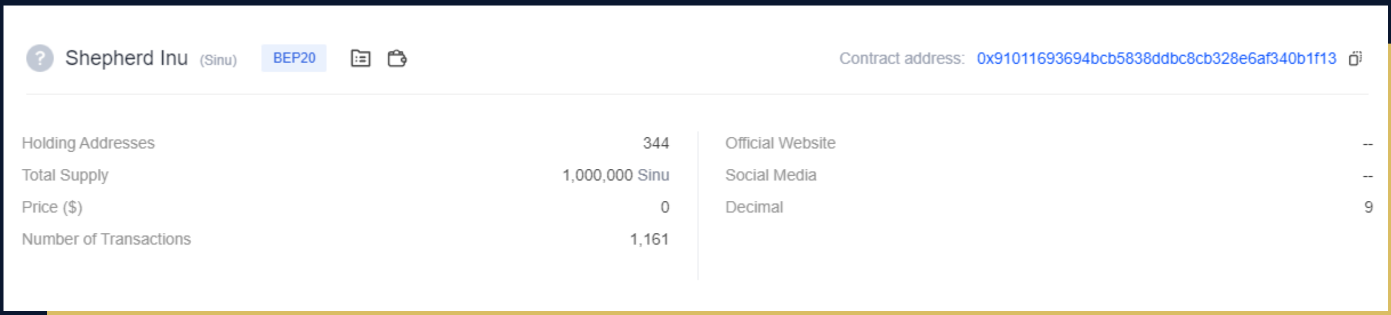
Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

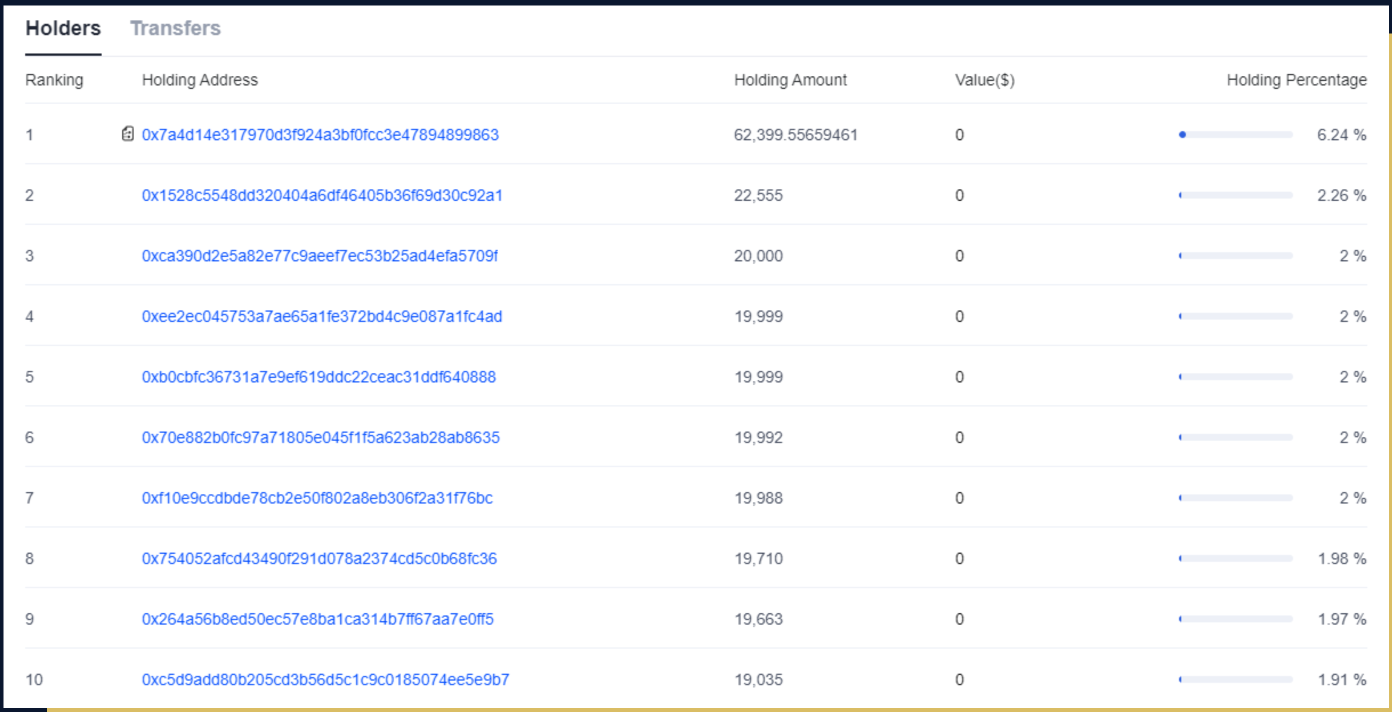
SHEPHERD INU TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



Shepherd Inu (Sinu) BEP20

Contract address: 0x91011693694bcb5838ddbc8cb328e6af340b1f13

Holding Addresses	344	Official Website	--
Total Supply	1,000,000 Sinu	Social Media	--
Price ($)	0	Decimal	9
Number of Transactions	1,161		



Holders Transfers

Ranking	Holding Address	Holding Amount	Value($)	Holding Percentage
1	0x7a4d14e317970d3f924a3bf0fcc3e47894899863	62,399.55659461	0	6.24 %
2	0x1528c5548dd320404a6df46405b36f69d30c92a1	22,555	0	2.26 %
3	0xca390d2e5a82e77c9aeef7ec53b25ad4efa5709f	20,000	0	2 %
4	0xee2ec045753a7ae65a1fe372bd4c9e087a1fc4ad	19,999	0	2 %
5	0xb0cbfc36731a7e9ef619ddc22ceac31ddf640888	19,999	0	2 %
6	0x70e882b0fc97a71805e045f1f5a623ab28ab8635	19,992	0	2 %
7	0xf10e9ccdbde78cb2e50f802a8eb306f2a31f76bc	19,988	0	2 %
8	0x754052afcd43490f291d078a2374cd5c0b68fc36	19,710	0	1.98 %
9	0x264a56b8ed50ec57e8ba1ca314b7ff67aa7e0ff5	19,663	0	1.97 %
10	0xc5d9add80b205cd3b56d5c1c9c0185074ee5e9b7	19,035	0	1.91 %

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.