freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Strelka AI

$STRELKA AI

22/02/2023

TOKEN OVERVIEW

Fees

- Buy fees: 7%
- Sell fees: 7%

Fees privileges

- Can change fees up to 10%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change / set max tx amount

Blacklist

- Blacklist function not detected

Other privileges

- Can exclude / include from fees

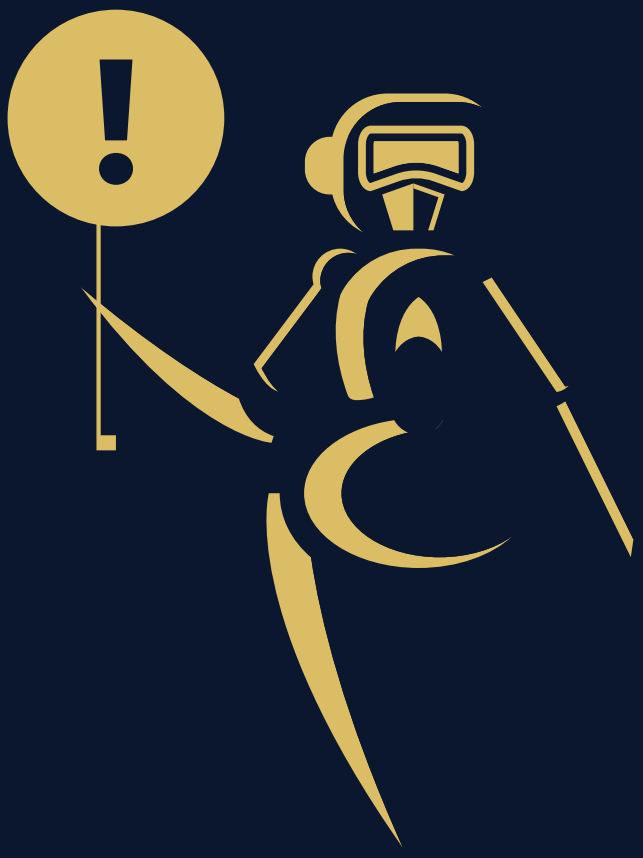
TABLE OF CONTENTS

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

0xab3BCB0E39b505de2a3545ce721E117DE75D1E1D

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 22/02/2023

AUDIT OVERVIEW

88

Security Score

98

Static Scan
Automatic scanning for common vulnerabilities

84

ERC Scan
Automatic checks for ERC's conformance

1 High

1 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't exclude an address from transactions
- Contract owner can exclude/include wallet from tx limitations

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

- Contract owner can change tx limitations (with threshold)

```
function setTxLimit(uint256 amount) external onlyOwner {
    require(amount > 4000);
    _maxTxAmount = amount * (10**_decimals);
}
```

- Contract owner can change fees up to 10%

```
function setTaxs(
    uint256 _reward,
    uint256 _liquidity,
    uint256 _operations,
    uint256 _marketing,
    uint256 _charity,
    uint256 _feeDenominator
) external onlyOwner {
    require(_feeDenominator<=10000, "Fee denominator can not be set over 100%");
    uint256 _total = _reward + _liquidity + _operations + _marketing + _charity;
    require(_total<=_feeDenominator/10, "Total tax can not be set over 10%"); /// Tax cannot exceed 10%

    taxRewardPool = _reward;
    taxLiquidity = _liquidity;
    taxOperations = _operations;
    taxMarketing = _marketing;
    taxCharity = _charity;
    feeDenominator = _feeDenominator;

    taxHandler.setRewardpoolTax(_reward);
    taxHandler.setLiquidityTax(_liquidity);
    taxHandler.setOperationsTax(_operations);
    taxHandler.setMarketingTax(_marketing);
    taxHandler.setCharityTax(_charity);
}
```

Contract owner can change rewardPool, LiquidityPool, OperationsPool, MarketingPool and CharityPool addresses

```
function setRewardPool(address _address) external onlyOwner {
    taxHandler.setRewardPool(_address);
}

function setLiquidityPool(address _address) external onlyOwner {
    taxHandler.setLiquidityPool(_address);
}

function setOperationsPool(address _address) external onlyOwner {
    taxHandler.setOperationsPool(_address);
}

function setMarketingPool(address _address) external onlyOwner {
    taxHandler.setMarketingPool(_address);
}

function setCharityPool(address _address) external onlyOwner {
    taxHandler.setCharityPool(_address);
}
```

Contract owner can change cooldown between trades settings

Trades can be disabled if cooldownTime is set to a big number

```
function setCooldownTime(uint256 _time) external onlyOwner {
    cooldownTime = _time;
}

function setMinPeriod(uint256 _minPeriod) external onlyOwner {
    taxHandler.setMinPeriod(_minPeriod);
}
```

Contract owner can transfer ownership

```
function transferOwnership(address payable adr) public onlyOwner {
    require(adr!=address(0));
    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 1 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	7%
Sell fees:	7%
Max TX:	100,000,000
Max Sell:	N/A

Honeypot Risk

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

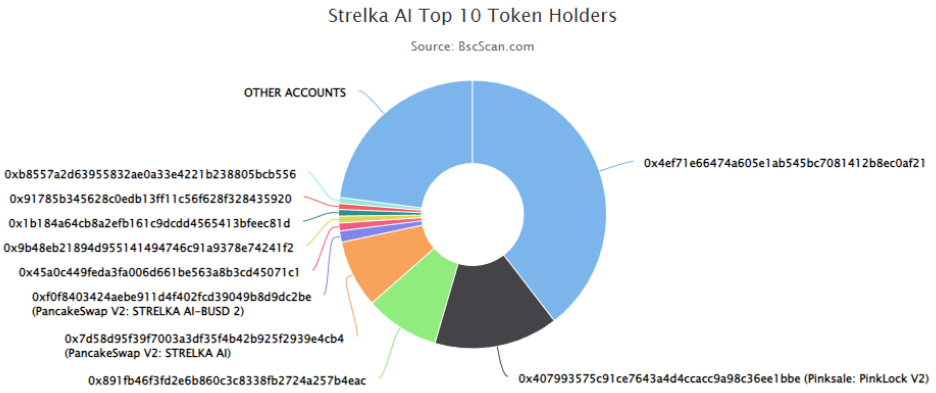
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

STRELKA AI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 77.05% (77,047,307.82 Tokens) of Strelka AI

Token Total Supply: 100,000,000.00 Token | Total Token Holders: 366



(A total of 77,047,307.82 tokens held by the top 10 accounts from the total supply of 100,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x4ef71e66474a605e1ab545bc7081412b8ec0af21	39,539,488.873712096251156495	39.5395%
2	Pinksale: PinkLock V2	15,000,000	15.0000%
3	0x891fb46f3fd2e6b860c3c8338fb2724a257b4eac	8,991,645.909583860905442441	8.9916%
4	PancakeSwap V2: STRELKA AI	8,151,895.631983148378896312	8.1519%
5	PancakeSwap V2: STRELKA AI-BUSD 2	1,372,900.171739249429420345	1.3729%
6	0x45a0c449feda3fa006d661be563a8b3cd45071c1	933,258.999390201202733058	0.9333%
7	0x9b48eb21894d955141494746c91a9378e74241f2	831,405.869628841591078433	0.8314%
8	0x1b184a64cb8a2efb161c9dcdd4565413bfeec81d	786,681.890843782328063231	0.7867%
9	0x91785b345628c0edb13ff11c56f628f328435920	720,015.23847229253081583	0.7200%
10	0xb8557a2d63955832ae0a33e4221b238805bcb556	720,015.23847229253081583	0.7200%

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.