freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Future Fox

$Helix

05/08/2023

TOKEN OVERVIEW

Fees

• Buy fees:	4%
• Sell fees:	8%

Fees privileges

- Can’t change / set fees

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can’t change max tx amount and / or max wallet amount

Blacklist

- Blacklist function not detected

Other privileges

- Can exclude / include from fees

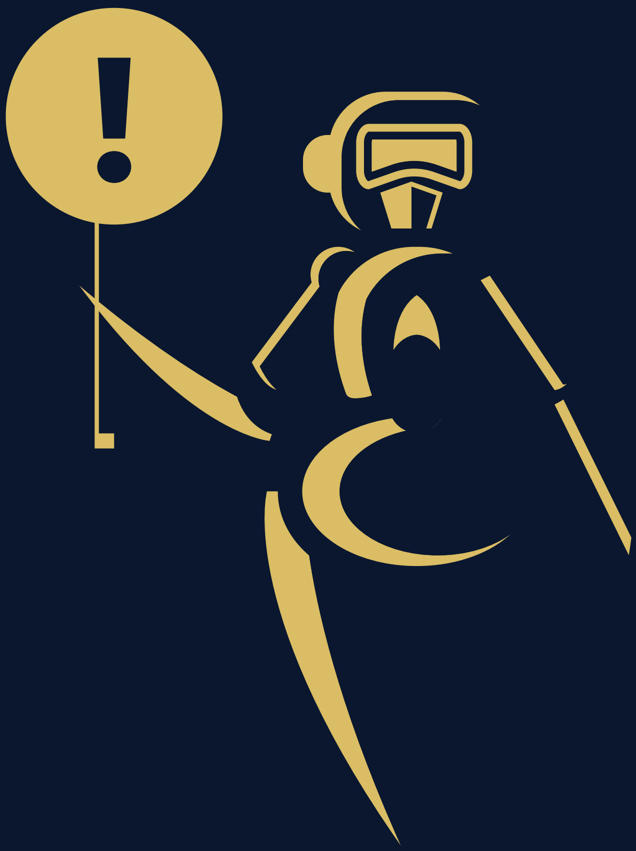
TABLE OF CONTENTS

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

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

0x6C6ae97DF635FFc7F0d81AfD46b70948dFb3E37f

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 05/08/2023

WEBSITE DIAGNOSTIC

https://futurefoxsite.com/

🟥	🟨	🟩
0-49	50-89	90-100

Performance	Accessibility	Best Practices	SEO	Progressive Web App
94	92	94	91	NA

Socials

Twitter

https://twitter.com/teamfuturefox

Telegram

https://t.me/futurefoxglobal

AUDIT OVERVIEW

94

Security Score

98

Static Scan

Automatic scanning for common vulnerabilities

80

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 mint tokens after initial contract deploy

- Contract owner can't exclude an address from transactions

- Contract owner can exclude/include wallet from tax

```
function setNoFeeWallet(address account, bool enabled) public onlyOwner {
    _noFee[account] = enabled;
}
```

- Contract owner can whitelist presale address

```
function setPresaleAddress(address presale, bool yesno) external onlyOwner {
        require(isPresaleAddress[presale] != yesno,"Same bool");
        isPresaleAddress[presale] = yesno;
        _noFee[presale] = yesno;
        liquidityAdd[presale] = yesno;
        emit _setPresaleAddress(presale, yesno);
}
```

- Contract owner can change buyFeeAddress, sellFeeAddress and mistakenlyAddress addresses

Default values:

buyFeeAddress: 0xcB2aed69240F4Ee654Ef39633B71A4792dFaD46B

sellFeeAddress: 0x6Fe143B540D81DE33ca042B670BE78855cD1B801

mistakenlyAddress: 0x2aDE143708B6BFDfC0fE2074575E5014806331Ae

```
function changeBuyFeeWallet(address newWallet) external onlyOwner {
    buyFeeAddress = payable(newWallet);
}

function changeSellFeeWallet(address newWallet) external onlyOwner {
    sellFeeAddress = payable(newWallet);
}

function changeMistakenlyWallet(address newWallet) external onlyOwner {
    mistakenlyAddress = payable(newWallet);
}
```

Contract owner can change swap settings

```
function toggleCanSwapFees(bool yesno) external onlyOwner {
        require(canSwapFees != yesno,"Bool is the same");
        canSwapFees = yesno;
        emit _toggleCanSwapFees(yesno);
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
        require(newOwner != address(0), "Ownable: new owner is the zero address");
        _setOwner(newOwner);
}

function _setOwner(address newOwner) private {
        address oldOwner = _owner;
        _owner = newOwner;
        emit OwnershipTransferred(oldOwner, newOwner);
}
```

Contract owner can renounce ownership

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

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 no HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	4%
Sell fees:	8%
Max TX:	N/A
Max Sell:	N/A

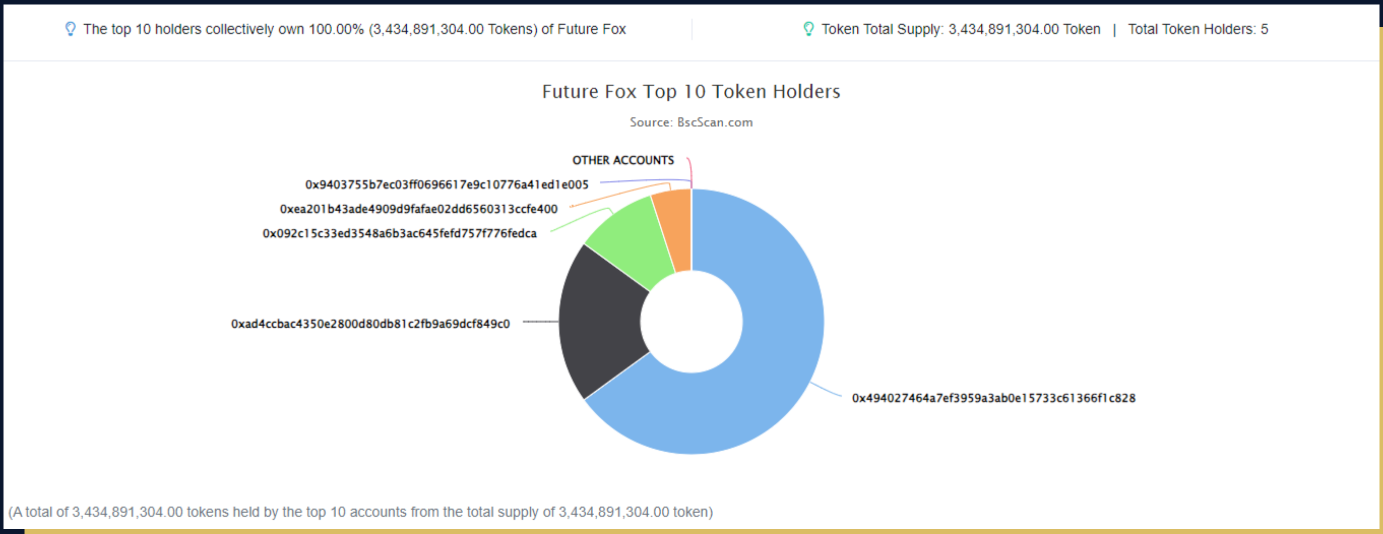
Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	100% unlocked tokens

HELIX TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



The top 10 holders collectively own 100.00% (3,434,891,304.00 Tokens) of Future Fox

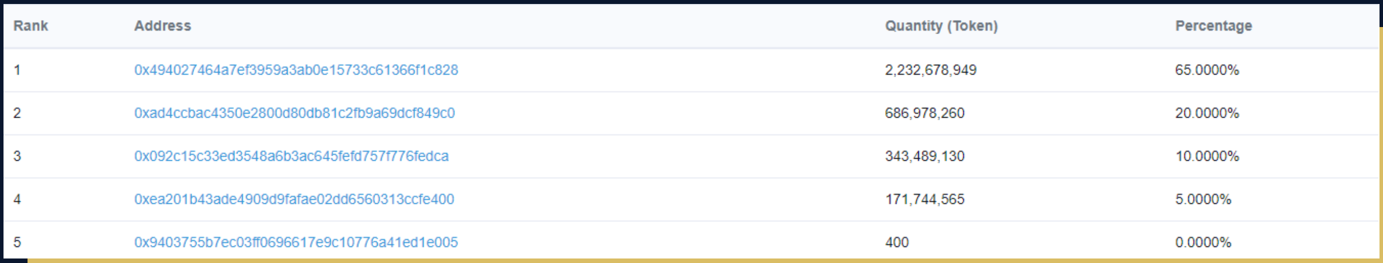
Token Total Supply: 3,434,891,304.00 Token | Total Token Holders: 5

Future Fox Top 10 Token Holders

Source: BscScan.com

OTHER ACCOUNTS
0x9403755b7ec03ff0696617e9c10776a41ed1e005
0xea201b43ade4909d9fafae02dd6560313ccfe400
0x092c15c33ed3548a6b3ac645fefd757f776fedca
0xad4ccbac4350e2800d80db81c2fb9a69dcf849c0
0x494027464a7ef3959a3ab0e15733c61366f1c828

(A total of 3,434,891,304.00 tokens held by the top 10 accounts from the total supply of 3,434,891,304.00 token)



Rank	Address	Quantity (Token)	Percentage
1	0x494027464a7ef3959a3ab0e15733c61366f1c828	2,232,678,949	65.0000%
2	0xad4ccbac4350e2800d80db81c2fb9a69dcf849c0	686,978,260	20.0000%
3	0x092c15c33ed3548a6b3ac645fefd757f776fedca	343,489,130	10.0000%
4	0xea201b43ade4909d9fafae02dd6560313ccfe400	171,744,565	5.0000%
5	0x9403755b7ec03ff0696617e9c10776a41ed1e005	400	0.0000%

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.