freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Esports Token

$ESPT

08/08/2022

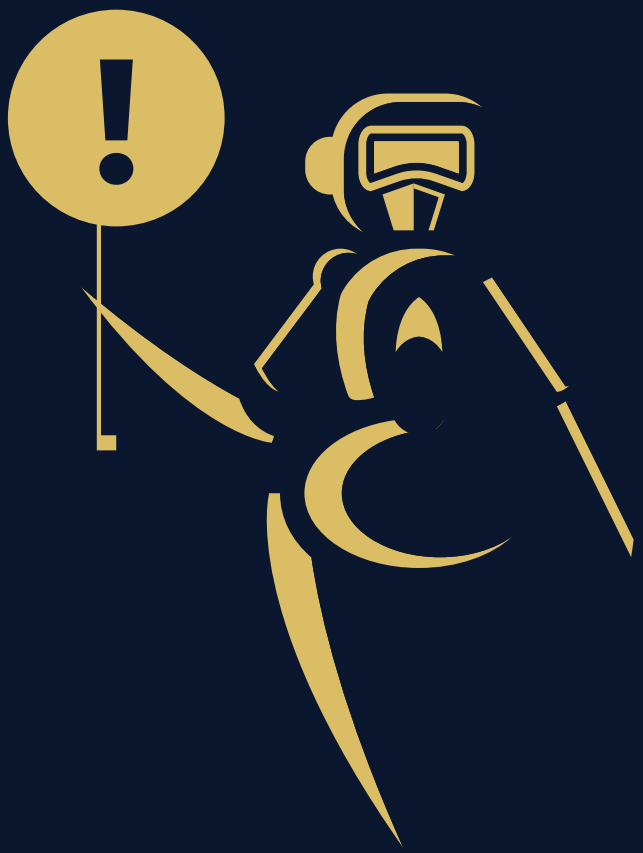
TABLE OF CONTENTS

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

0xE30a86A0b5aB2F47fFd37e9AD2DF1f8dA9419C30

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 08/08/2022

AUDIT OVERVIEW

66

Security Score

HIGH RISK

96

Static Scan

Automatic scanning for common vulnerabilities

40

ERC Scan

Automatic checks for ERC's conformance

4 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 mint tokens after initial contract deploy

```
function mint(uint256 amount) external onlyOperator {
    _mint(_msgSender(), amount);
}

function mint(address account, uint256 amount) external onlyOperator {
    _mint(account, amount);
}
```

Contract owner can exclude an address from transactions

```
function blacklistAccount(address account, bool flag) external onlyOwner {
    _blacklist[account] = flag;
}
```

Contract owner can burn tokens from a specific wallet

```
function burn(uint256 amount) external {
    _burn(_msgSender(), amount);
}

function _burn(address account, uint256 amount) internal virtual override {
    super._burn(account, amount);
    _afterTokenTransfer(account, address(0), amount);
}
```

Contract owner can exclude/include wallet from tax

```
function excludeFromTax(address account, bool flag) external onlyOwner {
    _excludedFromTax[account] = flag;
}
```

Contract owner can exclude/include wallet from tx limitations

```
function excludeFromTxLimit(address account, bool flag) external onlyOwner {
    _excludedFromTxLimit[account] = flag;
}
```

Contract owner can change max wallet amount (without threshold)

```
function excludeFromHoldLimit(address account, bool flag) external onlyOwner {
    _excludedFromHoldLimit[account] = flag;
}
```

Contract owner can enable/disable swap

```
function enableSwap(bool flag) external onlyOwner {
    _swapEnabled = flag;
}
```

Contract owner can change max tx amount and max wallet amount

```
uint256 public constant MAX_TX_AMOUNT_MIN_LIMIT = 100 ether;
uint256 public constant MAX_WALLET_AMOUNT_MIN_LIMIT = 1000 ether;
```

Current values:
_txLimit: 100000000000000000000000 (100,000 ESPT)
_holdLimit: 1000000000000000000000000 (10,000,000 ESPT)

```
function setAntiWhalesConfiguration(uint256 txLimit, uint256 holdLimit)
        external
        onlyOwner
    {
        require(txLimit >= MAX_TX_AMOUNT_MIN_LIMIT, "Max tx amount too small");
        require(
            holdLimit >= MAX_WALLET_AMOUNT_MIN_LIMIT,
            "Max wallet amount too small"
        );
        _txLimit = txLimit;
        _holdLimit = holdLimit;
    }
```

Contract owner can change _marketingWallet and _donationWallet addresses

Current values:
_marketingWallet : 0xd497564b9eef8b9f90f767d347db9cf0afee3a1b
_donationWallet : 0x45a4c5e3dbc784fa520c1918d31012cd59ee6dcd

```
function setMarketingWallet(address wallet) external onlyOwner {
        require(wallet != address(0), "Invalid marketing wallet");
        _marketingWallet = wallet;
    }

    function setDonationWallet(address wallet) external onlyOwner {
        require(wallet != address(0), "Invalid donation wallet");
        _donationWallet = wallet;
    }
```

Contract owner can change marketing token

Current values:
0xe9e7cea3dedca5984780bafc599bd69add087d56 (Binance-Peg BUSD Token (BUSD))

```
function setMarketingToken(address token) external onlyOwner {
        _marketingToken = token;
    }
```

Contract owner can change fees up to 100%

```
function setFee(
        TX_CASE feeCase,
        uint16 marketingFee,
        uint16 donationFee,
        uint16 burnFee
    ) external onlyOwner {
        require(marketingFee + donationFee + burnFee <= 10000, "Overflow");
        _tokenFees[feeCase].marketingFee = marketingFee;
        _tokenFees[feeCase].donationFee = donationFee;
        _tokenFees[feeCase].burnFee = burnFee;
    }
```

- 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;
}
```

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/operator.

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 4 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	10%
Sell fees:	10%
Max TX:	100,000
Max Sell:	N/A

Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

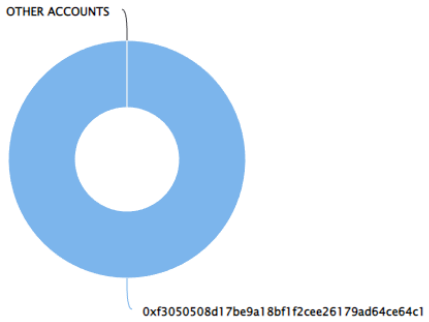
ESPORTS TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (6,000,000,000.00 Tokens) of Esports Token

Token Total Supply: 6,000,000,000.00 Token | Total Token Holders: 1

Esports Token Top 10 Token Holders

Source: BscScan.com



(A total of 6,000,000,000.00 tokens held by the top 10 accounts from the total supply of 6,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xf3050508d17be9a18bf1f2cee26179ad64ce64c1	6,000,000,000	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.