



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



SafePets

\$SFP

30/01/2022

TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 WEBSITE DIAGNOSTIC
- 5-6 AUDIT OVERVIEW
- 7-9 OWNER PRIVILEGES
- 10 CONCLUSION AND ANALYSIS
- 11 TOKEN DETAILS
- 12 SAFEPETS TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS
- 13 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 SafePets (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x7AA50494C6ce91f346660fd3a113D72B2E909572

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 30/01/2022



WEBSITE DIAGNOSTIC

<https://safepetstoken.com/>



0-49



50-89



90-100



Performance



Accessibility



Best Practices



SEO



Progressive
Web App

Metrics



First Contentful Paint

1.0 s



Time to interactive

4.3 s



Speed Index

1.8 s



Total Blocking Time

490 ms



Large Contentful Paint

1.6 s



Cumulative Layout Shift

0.001

WEBSITE IMPROVEMENTS

Reduce unused JavaScript

Ensure text remains visible during webfont load

Image elements do not have explicit `width` and `height`

Image elements do not have `[alt]` attributes

Background and foreground colors have a sufficient contrast ratio



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



ERC Scan
Automatic checks for ERC's conformance

 0 High

 0 Medium

 1 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	Low
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/include wallet from reward

```
function excludeFromReward(address account) public onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded from reward");
    if(_rOwned[account] > 0)
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

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;
}
```

Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

Contract owner can exclude/include wallet from fee

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

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

Contract owner can change max tx amount and wallet amount

```
uint8 public minMxTxPercentage = 1;
```

```
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    require(maxTxPercent >= minMxTxPercentage && maxTxPercent <= 10000, "err");
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**4
    );
}
```

```
uint8 public minMxWalletPercentage = 1;
```

```
function setMaxWalletPercent(uint256 maxWalletPercent) external onlyOwner() {
    require(maxWalletPercent >= minMxWalletPercentage && maxWalletPercent <= 10000, "err");
    _maxWalletAmount = _tTotal.mul(maxWalletPercent).div(
        10**4
    );
}
```

Contract owner can change `feeWallet` and `feeWalletCharity` address

Current addresses:

`feeWallet`: 0x66d8430569b615d0f66d8b4f4ad7707ddaf1d500

`feeWalletCharity`: 0x16fb5d7e4e68d788de1a16c5172ab1577d934915

```
function setFeeWallet(address payable newFeeWallet) external onlyOwner {
    require(newFeeWallet != address(0), "ZERO ADDRESS");
    feeWallet = newFeeWallet;
}
```

```
function setFeeWalletCharity(address payable newFeeWallet) external onlyOwner {
    require(newFeeWallet != address(0), "ZERO ADDRESS");
    feeWalletCharity = newFeeWallet;
}
```

Contract owner can change fees up to 50%

```
uint8 public maxLiqFee = 10;
uint8 public maxTaxFee = 10;
uint8 public maxBurnFee = 10;
uint8 public maxWalletFee = 10;
uint8 public maxBuybackFee = 10;

function setAllFeePercent(uint8 taxFee, uint8 liquidityFee, uint8 burnFee, uint8 walletFee, uint8 buybackFee,
uint8 walletCharityFee, uint8 rewardFee) external onlyOwner() {
    require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
    require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
    require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
    require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
    require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
    require(walletCharityFee >= 0 && walletCharityFee <=maxWalletFee,"WFT err");
    require(rewardFee >= 0 && rewardFee <=maxTaxFee,"RF err");
    //both tax fee and reward fee cannot be set
    require(rewardFee == 0 || taxFee == 0,"RT fee err");
    _taxFee = taxFee;
    _liquidityFee = liquidityFee;
    _burnFee = burnFee;
    _buybackFee = buybackFee;
    _walletFee = walletFee;
    _walletCharityFee = walletCharityFee;
    _rewardFee = rewardFee;
}
```

Contract owner is able to blacklist wallet(s)

```
function blacklistAddress(address account, bool value) external onlyOwner {
    _isBlacklisted[account] = value;
}
```

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 LOW issue during the first review.

TOKEN DETAILS

Details

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

Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



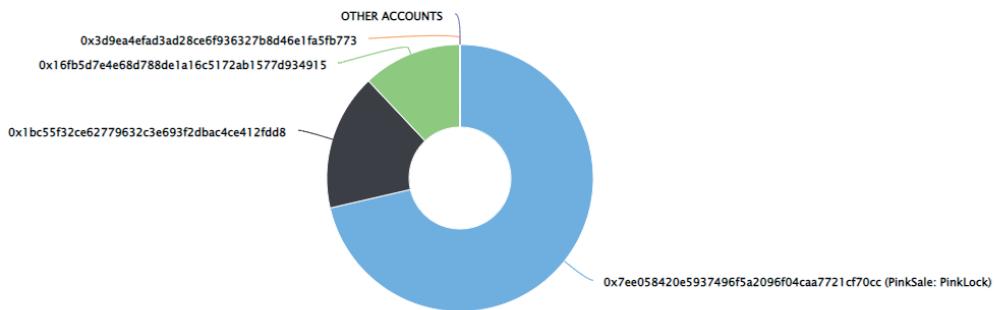
SAFEPETS TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000,000.00 Tokens) of SafePets

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 4

SafePets Top 10 Token Holders

Source: BscScan.com



(A total of 1,000,000,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x7ee058420e5937496f5a2096f04caa7721cf70cc (PinkSale: PinkLock)	714,000,000	71.4000%
2	0x1bc55f32ce62779632c3e693f2dbac4ce412fd8	165,699,834.3	16.5700%
3	0x16fb5d7e4e68d788de1a16c5172ab1577d934915	120,250,165.7	12.0250%
4	0x3d9ea4efad3ad28ce6f936327b8d46e1fa5fb773	50,000	0.0050%

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.

