Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

August, 2021

Audit Details

Audited project

MUSO Finance

Deployer address

0x627c95b6fd9026e00ab2c373fb08cc47e02629a0

Client contacts:

MUSO Finance team

Blockchain

Binance Smart Chain

Project website:

https://muso.finance

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by MUSO Finance to perform an audit of smart contracts:

https://bscscan.com/address/0x20512ee0052236b009772af0ed22bc58b40c27b9#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

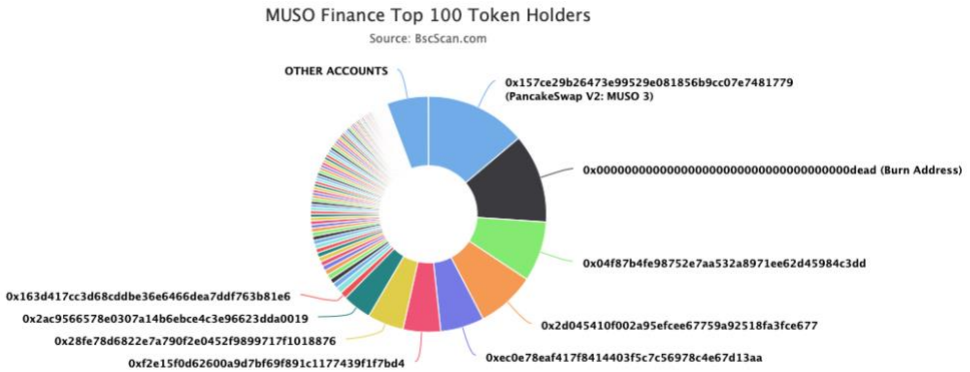
Contracts Details

Token contract details for 19.08.2021

Contract name	MUSO Finance
Contract address	0x20512Ee0052236B009772Af0Ed22BC58B40c27B9
Total supply	1,000,000,000,000,000
Token ticker	MUSO
Decimals	9
Token holders	1,000
Transactions count	3,494
Top 100 holders dominance	94.28%
Reflection fee	4
Tax fee	3
Total fees	22893117607772439321833
Uniswap V2 pair	0x157ce29b26473e99529e081856b9cc07e7481779
Contract deployer address	0x627c95b6fd9026e00ab2c373fb08cc47e02629a0
Contract's current owner address	0x04f87b4fe98752e7aa532a8971ee62d45984c3dd

MUSO Finance Token Distribution

The top 100 holders collectively own 94.28% (942,805,654,640,540.00 Tokens) of MUSO Finance | Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 1,000

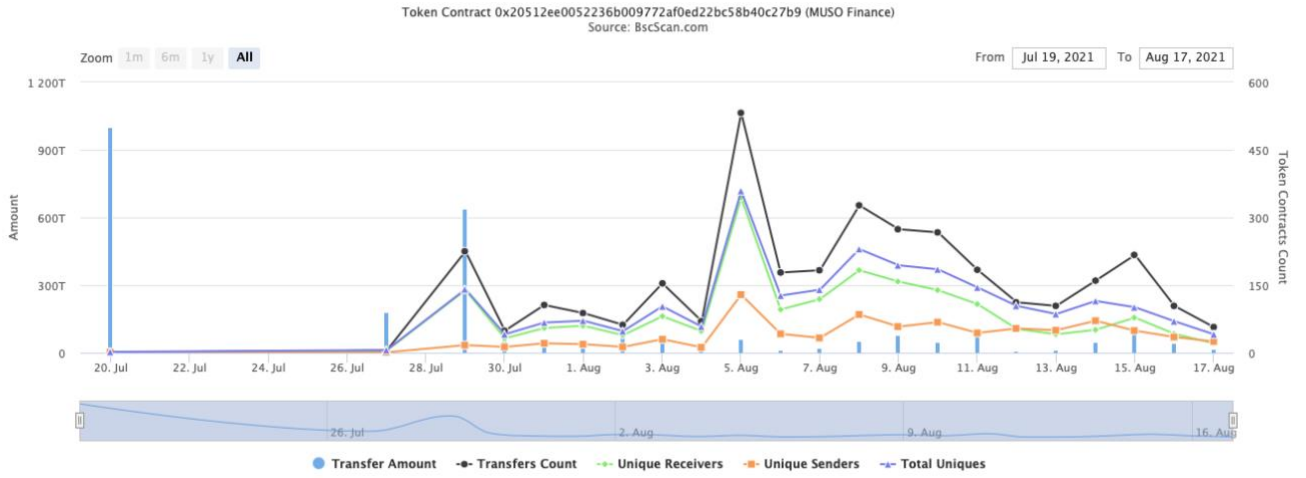


(A total of 942,805,654,640,540.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

MUSO Finance Contract Interaction Details

Time Series: Token Contract Overview

Tue 20, Jul 2021 - Tue 17, Aug 2021



MUSO Finance Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: MUSO 3	138,670,058,540,729.744765259	13.8670%
2	Burn Address	121,944,607,277,032.775021564	12.1945%
3	0x04f87b4fe98752e7aa532a8971ee62d45984c3dd	82,496,881,999,807.308247935	8.2497%
4	0x2d045410f002a95efcee67759a92518fa3fce677	80,330,810,731,257.013678696	8.0331%
5	0xec0e78eaf417f8414403f5c7c56978c4e67d13aa	60,000,000,000,000	6.0000%
6	0xf2e15f0d62600a9d7bf69f891c1177439f1f7bd4	51,505,254,252,779.568825426	5.1505%
7	0x28fe78d6822e7a790f2e0452f9899717f1018876	49,611,878,298,416.816500849	4.9612%
8	0x2ac9566578e0307a14b6ebce4c3e96623dda0019	40,000,000,000,000	4.0000%
9	0x163d417cc3d68cddbe36e6466dea7ddf763b81e6	9,123,752,081,460.00720119	0.9124%
10	0xa66800b4cca86a26d6096a5e2eb0784205aedcb8	8,783,480,787,265.279223504	0.8783%

MUSO Finance LP Token Holders

Rank	Address	Quantity	Percentage
1	0x8655e5c4d701186d16765d1cdcef6d5287e4679a	7,252.419344911696802766	89.1791%
2	0x04f87b4fe98752e7aa532a8971ee62d45984c3dd	876.956502536143550507	10.7835%
3	0x8cc7bc33f5188b1fb683bedc4dbffa77b136833b	3.045833759433563027	0.0375%
4	0x0000000000000000000000000000000000000000	0.000000000000001	0.0000%

Contract functions details

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] sub
  - [Int] div

+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Int] functionStaticCall
  - [Int] functionStaticCall
  - [Int] functionDelegateCall #
  - [Int] functionDelegateCall #
  - [Prv] _verifyCallResult

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner

+ [Int] IUniswapV2Factory
  - [Ext] feeTo
  - [Ext] feeToSetter
  - [Ext] getPair
  - [Ext] allPairs
  - [Ext] allPairsLength
  - [Ext] createPair #
  - [Ext] setFeeTo #
  - [Ext] setFeeToSetter #

+ [Int] IUniswapV2Pair
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transfer #
  - [Ext] transferFrom #
  - [Ext] DOMAIN_SEPARATOR
  - [Ext] PERMIT_TYPEHASH
  - [Ext] nonces
  - [Ext] permit #
  - [Ext] MINIMUM_LIQUIDITY
  - [Ext] factory
  - [Ext] token0
  - [Ext] token1
  - [Ext] getReserves
  - [Ext] price0CumulativeLast
  - [Ext] price1CumulativeLast
  - [Ext] kLast
  - [Ext] mint #
  - [Ext] burn #
  - [Ext] swap #
  - [Ext] skim #
  - [Ext] sync #
  - [Ext] initialize #

+ [Int] IUniswapV2Router01
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidity #
  - [Ext] addLiquidityETH ($)
  - [Ext] removeLiquidity #
  - [Ext] removeLiquidityETH #
  - [Ext] removeLiquidityWithPermit #
  - [Ext] removeLiquidityETHWithPermit #
  - [Ext] swapExactTokensForTokens #
  - [Ext] swapTokensForExactTokens #
  - [Ext] swapExactETHForTokens ($)
  - [Ext] swapTokensForExactETH #
  - [Ext] swapExactTokensForETH #
  - [Ext] swapETHForExactTokens ($)
  - [Ext] quote
  - [Ext] getAmountOut
  - [Ext] getAmountIn
  - [Ext] getAmountsOut
  - [Ext] getAmountsIn

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
  - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
  - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
  - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
  - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)

- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ MUSO_Finance (Context, IERC20, Ownable)
- [Pub] <Constructor> #
- [Pub] name
- [Pub] symbol
- [Pub] decimals
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer #
- [Pub] allowance
- [Pub] approve #
- [Pub] transferFrom #
- [Pub] increaseAllowance #
- [Pub] decreaseAllowance #
- [Pub] isExcludedFromReward
- [Pub] totalFees
- [Pub] sefeLaunch_buy_buy_delay #
  - modifiers: onlyOwner
- [Pub] sefeLaunch_buy_sell_delay #
  - modifiers: onlyOwner
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward #
  - modifiers: onlyOwner
- [Ext] includeInReward #
  - modifiers: onlyOwner
- [Prv] _transferBothExcluded #
- [Ext] __prepareForPreSale #
  - modifiers: onlyOwner
- [Ext] __afterPreSale #
  - modifiers: onlyOwner
- [Pub] safeLaunch_setOnlyCommunity #
  - modifiers: onlyOwner
- [Pub] excludeFromFee #
  - modifiers: onlyOwner
- [Pub] includeInFee #
  - modifiers: onlyOwner
- [Ext] process_setNumTokensSellToAddToLiquidity #
  - modifiers: onlyOwner
- [Ext] fees_setFeeReflectionPercent #
  - modifiers: onlyOwner
- [Ext] fees_setFeeTokenGiveawayPercent #
  - modifiers: onlyOwner
- [Ext] fees_setFeeLiquidityPercent #
  - modifiers: onlyOwner
- [Ext] fees_setFeePromoPercent #
  - modifiers: onlyOwner
- [Ext] safeLaunch_setMaxTxPercent #
  - modifiers: onlyOwner
- [Ext] safeLaunch_setMaxTxTokens #
  - modifiers: onlyOwner
- [Ext] safeLaunch_setMaxWalletPercent #
  - modifiers: onlyOwner

- [Ext] safeLaunch_setMaxWalletTokens #
  - modifiers: onlyOwner
- [Pub] process_setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Ext] <Fallback> ($)
- [Prv] _reflectFee #
- [Ext] safeLaunch_removeFromWhitelist #
  - modifiers: onlyOwner
- [Ext] safeLaunch_removeFromBlackList #
  - modifiers: onlyOwner
- [Ext] safeLaunch_addToWhitelist #
  - modifiers: onlyOwner
- [Ext] safeLaunch_addToBlackList #
  - modifiers: onlyOwner
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [Prv] _takeDev #
- [Prv] calculateReflectionFee
- [Prv] calculateTokenGiveawayFee
- [Prv] calculateLiquidityAndPromoFee
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Pub] isExcludedFromFee
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] sendToPromoWallet #
- [Prv] swapAndLiquify #
  - modifiers: lockTheSwap
- [Prv] swapTokensForEth #
- [Prv] addLiquidity #
- [Pub] process_TokensFromContract #
  - modifiers: onlyOwner
- [Pub] process_BNBFromContract #
  - modifiers: onlyOwner
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #

($) = payable function
= non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Low issues
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Passed
19.	Cross-function race conditions.	Passed
20.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Out of gas

Issue:

- The function includeInReward() uses the loop to find and remove addresses from the _excluded list. Function will be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
function includeInReward(address account⬆) external onlyOwner() {
    require(_isExcluded[account⬆], "Account is 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;
        }
    }
}
```

- The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
        ) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

- Functions safeLaunch_removeFromWhitelist(), safeLaunch_removeFromBlackList(), safeLaunch_addToWhitelist(), safeLaunch_addToBlackList() also uses the loop for

whitelist/blacklist addresses. Them also could be aborted with OUT_OF_GAS exception if there will be a long addresses list.

```
//Remove from whitelist
  ftrace | funcSig
  function safeLaunch_removeFromWhitelist(address[] calldata addresses⬆) external onlyOwner {
  for (uint256 i; i < addresses⬆.length; ++i) {
    _isCommunity[addresses⬆[i]] = false;
  }
}

//Remove from Blacklist
 ftrace | funcSig
 function safeLaunch_removeFromBlackList(address[] calldata addresses⬆) external onlyOwner {
  for (uint256 i; i < addresses⬆.length; ++i) {
    _isBlacklisted[addresses⬆[i]] = false;
  }
}


//adding people to the whitelist - these people are the only ones that will be able to buy at launch!
ftrace | funcSig
function safeLaunch_addToWhitelist(address[] calldata addresses⬆) external onlyOwner {
  for (uint256 i; i < addresses⬆.length; ++i) {
    _isCommunity[addresses⬆[i]] = true;
  }
}

//adding multiple addresses to the blacklist - Used to manually block known bots and scammers
ftrace | funcSig
function safeLaunch_addToBlackList(address[] calldata addresses⬆) external onlyOwner {
  for (uint256 i; i < addresses⬆.length; ++i) {
    _isBlacklisted[addresses⬆[i]] = true;
  }
}
```

Recommendation:

Check that the arrays length is not too big.

Owner privileges (In the period when the owner is not renounced)

- Owner can change fees.

```
//set the fee that is automatically distributed to all holders (reflection)
ftrace | funcSig
function fees_setFeeReflectionPercent(uint256 reflectionFee⬆) external onlyOwner() {
    _reflectionFee = reflectionFee⬆;
}

//set fee for the giveaway and manual burn wallet
ftrace | funcSig
function fees_setFeeTokenGiveawayPercent(uint256 tokenGiveawayFee⬆) external onlyOwner() {
    _tokenGiveawayFee = tokenGiveawayFee⬆;
}

//set fee for auto liquidity
ftrace | funcSig
function fees_setFeeLiquidityPercent(uint256 liquidityFee⬆) external onlyOwner() {
    _liquidityFee = liquidityFee⬆;
}

//set fee for the marketing (BNB) wallet
ftrace | funcSig
function fees_setFeePromoPercent(uint256 promoFee⬆) external onlyOwner() {
    _promoFee = promoFee⬆;
}
```

- Owner can change buy and sell delays.

```
ftrace | funcSig
function sefeLaunch_buy_buy_delay(bool setBool⬆, uint8 numSeconds⬆) public onlyOwner {
    slowFairBuys = setBool⬆;
    buy_buy_delay = numSeconds⬆;
}

//delay from a buy to a sell in seconds
ftrace | funcSig
function sefeLaunch_buy_sell_delay(bool setBool⬆, uint8 numSeconds⬆) public onlyOwner {
    slowFairBuys = setBool⬆;
    buy_sell_delay = numSeconds⬆;
}
```

- Owner can change the maximum transaction amount.

```
//set the Max transaction amount (percent of total supply)
ftrace | funcSig
function safeLaunch_setMaxTxPercent(uint256 maxTxPercent⬆) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent⬆).div(
        10**2
    );
}

//set the Max transaction amount (in tokens)
 ftrace | funcSig
 function safeLaunch_setMaxTxTokens(uint256 maxTxTokens⬆) external onlyOwner() {
    _maxTxAmount = maxTxTokens⬆;
}
```

- Owner can exclude from the fee.

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

- Owner can activate prepare and after presale modes.

```
//get ready for presale!
   ftrace | funcSig
   function __prepareForPreSale() external onlyOwner {
     process_setSwapAndLiquifyEnabled(false);
     _reflectionFee = 0;
     _liquidityFee = 0;
     _promoFee = 0;
     _liquidityAndPromoFee = _liquidityFee+_promoFee;
     _maxTxAmount = 1000000000 * 10**6 * 10**9;
     _maxWalletToken = 1000000000 * 10**6 * 10**9;
}

//presale done! Set all fees
ftrace | funcSig
function __afterPreSale() external onlyOwner {
    process_setSwapAndLiquifyEnabled(true);
    _reflectionFee = 3;
    _liquidityFee = 4;
    _promoFee = 3;
    _liquidityAndPromoFee = _liquidityFee+_promoFee;
    _maxTxAmount = 10000000 * 10**6 * 10**9;
    _maxWalletToken = 20000000 * 10**6 * 10**9;
}
```

- Owner can enable/disable community mode.

```
function safeLaunch_setOnlyCommunity(bool _enabled⬆) public onlyOwner {
    onlyCommunity = _enabled⬆;
}
```

- Owner can change number of tokens to sell to add to liquidity.

```
function process_setNumTokensSellToAddToLiquidity(uint256 numTokensSellToAddToLiquidity⬆) external onlyOwner() {
    _numTokensSellToAddToLiquidity = numTokensSellToAddToLiquidity⬆;
}
```

- Owner can change max wallet token.

```
//settting the maximum permitted wallet holding (percent of total supply)
 ftrace | funcSig
 function safeLaunch_setMaxWalletPercent(uint256 maxWallPercent⬆) external onlyOwner() {
    _maxWalletToken = _tTotal.mul(maxWallPercent⬆).div(
        10**2
    );
}

//settting the maximum permitted wallet holding (in tokens)
 ftrace | funcSig
 function safeLaunch_setMaxWalletTokens(uint256 maxWallTokens⬆) external onlyOwner() {
    _maxWalletToken = maxWallTokens⬆;
}
```

- Owner can change max wallet token.

```
ftrace | funcSig
function safeLaunch_setMaxWalletPercent(uint256 maxWallPercent⬆) external onlyOwner() {
    _maxWalletToken = _tTotal.mul(maxWallPercent⬆).div(
        10**2
    );
}
```

- Owner can manually swap tokens to BNB and send them to promo wallet.

```
function process_TokensFromContract(uint256 tokenAmount⬆) public onlyOwner {
    uint256 tokensOnWallet = balanceOf(address(this));
    if (tokenAmount⬆ > tokensOnWallet) {tokenAmount⬆ = tokensOnWallet;}
    uint256 balanceBefore = address(this).balance;
    swapTokensForEth(tokenAmount⬆);
    uint256 balanceToSend = address(this).balance - balanceBefore;
    sendToPromoWallet(balanceToSend);
}
```

- Owner can send contract BNBs to promo wallet.

```
function process_BNBFromContract(uint256 bnbAmount⬆) public onlyOwner {
    uint256 contractBNB = address(this).balance;
    if (contractBNB > 0) {
    if (bnbAmount⬆ > contractBNB) {bnbAmount⬆ = contractBNB;}
    sendToPromoWallet(bnbAmount⬆);
}
}
```

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details provided by the team:
http://dxsale.app/app/v2_9/dxlockview?id=0&add=0x04f87B4Fe98752e7aa532A8971ee62D45984c3DD&type=lplock&chain=BSC

TechRate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.