Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

October, 2021

Audit Details

Audited project

Crosswise Token

Deployer address

0x6973a5D5e2Bd3bBDe498104FeCDF3132A3c545aB

Client contacts:

Crosswise Token team

Blockchain

Binance Smart Chain

Project website:

Not provided

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

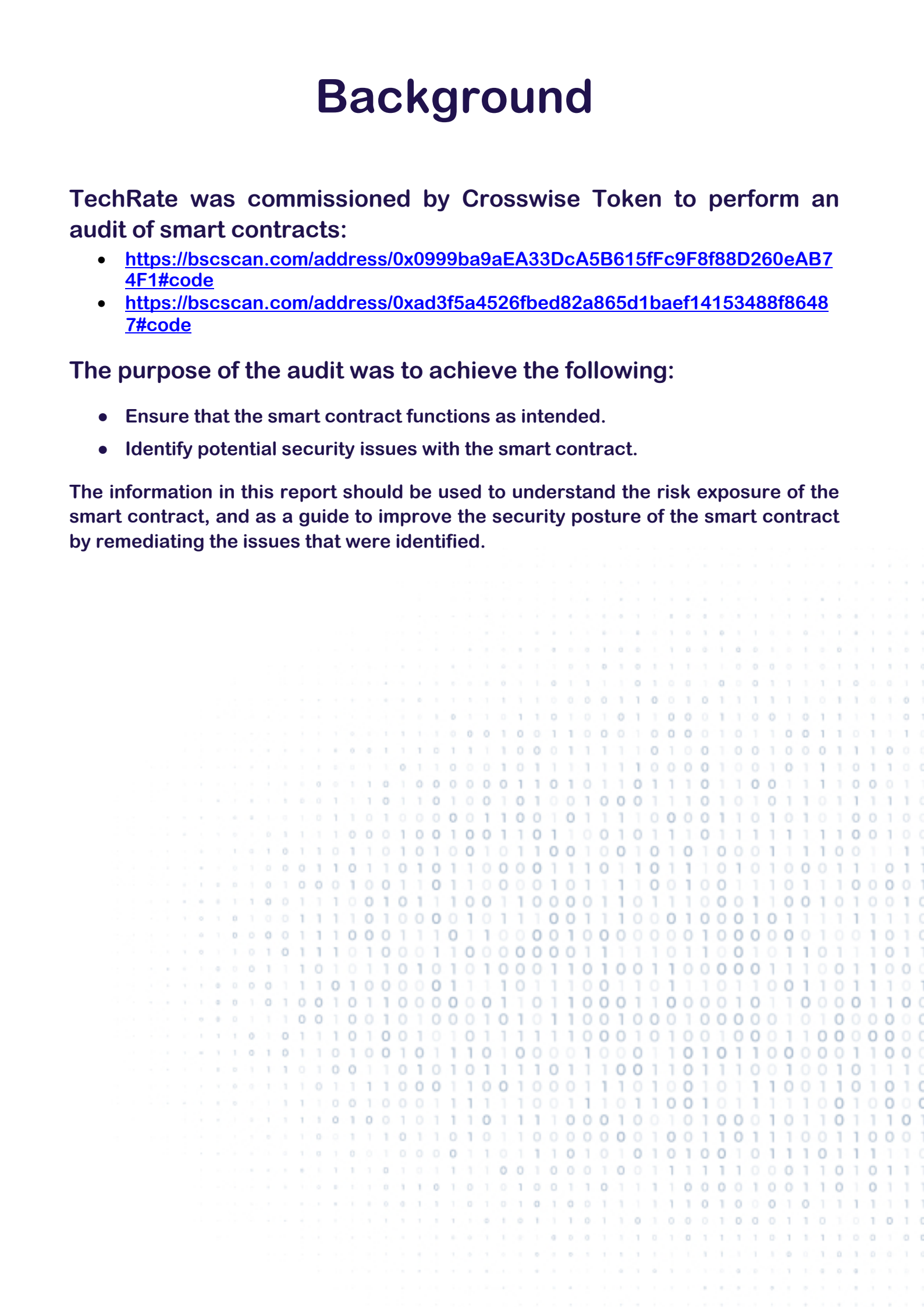
TechRate was commissioned by Crosswise Token to perform an audit of smart contracts:

- https://bscscan.com/address/0x0999ba9aEA33DcA5B615fFc9F8f88D260eAB74F1#code
- https://bscscan.com/address/0xad3f5a4526fbed82a865d1baef14153488f86487#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 24.10.2021

Contract name	Crosswise Token
Contract address	0x0999ba9aEA33DcA5B615fFc9F8f88D260eAB74F1
Total supply	0
Token ticker	CRSS
Decimals	18
Token holders	0
Transactions count	0
Top 100 holders dominance	100.00%
Contract deployer address	0x6973a5D5e2Bd3bBDe498104FeCDF3132A3c545aB
Contract’s current owner address	0x6973a5D5e2Bd3bBDe498104FeCDF3132A3c545aB

Contract functions details

+ [Int] ICrosswiseRouter02 (ICrosswiseRouter01)
  - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
  - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
  - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
  - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ [Int] ICrosswiseRouter01
  - [Ext] factory
  - [Ext] WBNB
  - [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] ICrosswiseFactory
  - [Ext] feeTo
  - [Ext] feeToSetter
  - [Ext] getPair
  - [Ext] allPairs
  - [Ext] allPairsLength
  - [Ext] createPair #
  - [Ext] setFeeTo #
  - [Ext] setFeeToSetter #

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

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

+ [Int] IBEP20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Lib] SafeMath
  - [Int] tryAdd
  - [Int] trySub
  - [Int] tryMul
  - [Int] tryDiv
  - [Int] tryMod
  - [Int] add
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] mod
  - [Int] sub
  - [Int] div
  - [Int] mod

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

+ CrssToken (Context, IBEP20, Ownable)
  - [Ext] <Fallback> ($)
  - [Pub] <Constructor> #
  - [Pub] init_router #
    - modifiers: onlyOwner
  - [Ext] getOwner
  - [Pub] name
  - [Pub] decimals
  - [Pub] symbol
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Int] _mint #
  - [Int] _approve #
  - [Pub] transfer #
    - modifiers: antiWhale

- [Pub] transferFrom #
  - modifiers: antiWhale
- [Int] _transfer #
- [Pub] setPresaleEnabled #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Pub] setMaxTransferAmountRate #
  - modifiers: onlyOwner
- [Prv] swapAndLiquify #
  - modifiers: lockTheSwap
- [Prv] swapTokensForEth #
- [Prv] addLiquidity #
- [Pub] mint #
  - modifiers: onlyOwner
- [Ext] delegates
- [Ext] delegate #
- [Ext] delegateBySig #
- [Ext] getCurrentVotes
- [Ext] getPriorVotes
- [Int] _delegate #
- [Int] _moveDelegates #
- [Int] _writeCheckpoint #
- [Pub] maxTransferAmount
- [Pub] isExcludedFromAntiWhale
- [Pub] setExcludedFromAntiWhale #
  - modifiers: onlyOwner
- [Int] safe32
- [Int] getChainId

($) = 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.	Passed
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

No low severity issues found.

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

Crosswise Token

- Owner can change router address.

```
function init_router(address router) public onlyOwner {
    ICrosswiseRouter02 _crosswiseRouter = ICrosswiseRouter02(router);
    // Create a uniswap pair for this new token
    crssBnbPair = ICrosswiseFactory(_crosswiseRouter.factory())
    .createPair(address(this), _crosswiseRouter.WBNB());

    // set the rest of the contract variables
    crosswiseRouter = _crosswiseRouter;
}
```

- Owner can enable / disable presale.

```
function setPresaleEnabled(bool _presaleEnabled) public onlyOwner {
    presaleEnabled = _presaleEnabled;
    emit PresaleEnabledUpdated(_presaleEnabled);
}
```

- Owner can enable / disable swap and liquify.

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

- Owner can change transfer max amount.

```
 function setMaxTransferAmountRate(uint256 _maxTransferAmountRate) public onlyOwner {
    require(_maxTransferAmountRate <= 10000,
    "CrssToken.setMaxTransferAmountRate: Max transfer amount rate must not exceed the maximum rate.");
    maxTransferAmountRate = _maxTransferAmountRate;
}
```

- Owner can mint tokens up to max supply amount.

```
function mint(address _to, uint256 _amount) public onlyOwner {
    _mint(_to, _amount);
    _moveDelegates(address(0), _delegates[_to], _amount);
}
```

- Owner can include in and exclude from anti whale.

```
function setExcludedFromAntiWhale(address _account, bool _excluded) public onlyOwner {
    _excludedFromAntiWhale[_account] = _excluded;
}
```

PreSale

- Owner can update soft cap, hard cap and second round amounts.

```
/**
 * @notice Method for updating softcap amount
 * @dev Only admin
 * @param _softCapAmount New softcap amount
 */
function updateSoftCapAmount(uint256 _softCapAmount) external onlyOwner {
    require(_softCapAmount > 0, "Presale.updateSoftCapAmount: soft cap amount invalid");
    softCapAmount = _softCapAmount;

    emit UpdateSoftCapAmount(_softCapAmount);
}

/**
 * @notice Method for updating hardcap amount
 * @dev Only admin
 * @param _hardCapAmount New hardcap amount
 */
function updateHardCapAmount(uint256 _hardCapAmount) external onlyOwner {
    require(_hardCapAmount > 0, "Presale.updateHardCapAmount: hard cap amount invalid");
    hardCapAmount = _hardCapAmount;

    emit UpdateHardCapAmount(_hardCapAmount);
}

/**
 * @notice Method for updating second round amount
 * @dev Only admin
 * @param _secondRoundAmount New second round amount
 */
function updateSecondRoundAmount(uint256 _secondRoundAmount) external onlyOwner {
    require(_secondRoundAmount > 0, "Presale.updateSoftCapAmount: soft cap amount invalid");
    secondRoundAmount = _secondRoundAmount;

    emit UpdateSoftCapAmount(_secondRoundAmount);
}
```

- Owner can update minimum purchase amount.

```
/**
 * @notice Method for updating min purchase
 * @dev Only admin
 * @param _minPurchase New min purchase per user
 */
function updateMinPurchase(uint256 _minPurchase) external onlyOwner {
    require(_minPurchase > 0, "Presale.updateMinPurchase: min purchase amount invalid");
    minPurchase = _minPurchase;

    emit UpdateMinPurchase(_minPurchase);
}
```

- Owner can update maximum amount per wallet.

```
/**
 * @notice Method for updating maxBusdPerWallet
 * @dev Only admin
 * @param _maxBusdPerWallet New maxBusdPerWallet
 */
function updateMaxBusdPerWallet(uint256 _maxBusdPerWallet) external onlyOwner {
    require(_maxBusdPerWallet > 0, "Presale.updateMaxBusdPerWallet: max busd amount invalid");
    maxBusdPerWallet = _maxBusdPerWallet;

    emit UpdateMaxBusdPerWallet(_maxBusdPerWallet);
}
```

- Owner can include in and exclude from deposit whitelist.

```
/**
 * @notice Method for setting whitelist address for presale
 * @dev Only admin
 * @param _addr Address for whitelist
 * @param _status Boolean value that determines to set/unset address to whitelist
 */
function setWhiteList(address _addr, bool _status) external onlyOwner {
    require(_addr != address(0), "Presale.setWhiteList: Zero Address");
    whitelist[_addr] = _status;

    emit SetWhiteList(_addr, _status);
}
```

Conclusion

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

Liquidity locking details NOT provided by the team.

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.