Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

September, 2021

Audit Details

Audited project

Spectra

Deployer address

0x52301120e79011ee21934d64c96be306c4336a8e

Client contacts:

Spectra team

Blockchain

Binance Smart Chain

Project website:

https://spectratoken.com

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 Spectra to perform an audit of smart contracts:

https://bscscan.com/address/0xa2f017966d967ec697C7A20Cf9D0b43CB8d4fF1D#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 08.09.2021

Contract name	Spectra
Contract address	0xa2f017966d967ec697C7A20Cf9D0b43CB8d4fF1D
Total supply	1,000,000,000
Token ticker	SPC
Decimals	9
Token holders	479
Transactions count	4,613
Top 100 holders dominance	95.56%
B/S Liquidity fee	0/5
Tax fee	0
Total fees	0
Uniswap V2 pair	0x6401f38e5d8ab79c7cf38e478982e7c410e70e3c
Contract deployer address	0x52301120e79011ee21934d64c96be306c4336a8e
Contract's current owner address	0x52301120e79011ee21934d64c96be306c4336a8e

Spectra Token Distribution

The top 100 holders collectively own 95.56% (955,646,014.91 Tokens) of Spectra

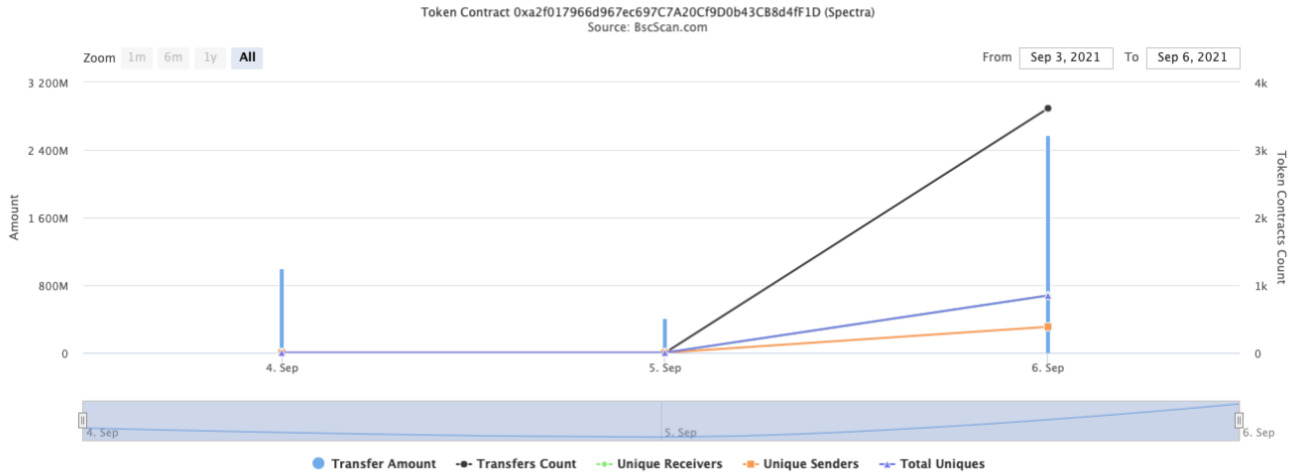
Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 479

(A total of 955,646,014.91 tokens held by the top 100 accounts from the total supply of 1,000,000,000.00 token)

Spectra Contract Interaction Details

Time Series: Token Contract Overview

Sat 4, Sept 2021 - Mon 6, Sept 2021



Spectra Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: SPC 111	135,764,102.245186516	13.5764%
2	0xa275c65e5e5adcdc15c7ed3841de833afe15a03c	103,366,241.11870331	10.3366%
3	0xa2f017966d967ec697c7a20cf9d0b43cb8d4ff1d	84,478,072.919654522	8.4478%
4	0x52301120e79011ee21934d64c96be306c4336a8e	80,970,624.944142101	8.0971%
5	0x3c5701eeae9f9f1dbe4ae83d71b685b7833fe2f3	78,356,036.728210293	7.8356%
6	0x389a6ac3c4d00625d9fc92d9c12fc42609ea9ca4	61,996,144.016136193	6.1996%
7	0xb5fccce093462890f3d388fa8c177f5973487f59	19,999,500.934947536	2.0000%
8	0x2a8d9d2d47c2fa97a05eb11154acf8b92c6f250b	19,956,002.026654578	1.9956%
9	0x036fde98491ac73eaca4c2303ef360a9100f2412	15,359,753.436649698	1.5360%
10	0xe1dcb2f12c99f74f3f1a8d1e215052b087f91514	15,322,637.792268355	1.5323%

Contract functions details

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

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

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

+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Prv] _functionCallWithValue #

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

+ [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] 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 #

+ Spectra (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] minimumTokensBeforeSwapAmount
- [Pub] buyBackUpperLimitAmount
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward #
  - modifiers: onlyOwner
- [Ext] includeInReward #
  - modifiers: onlyOwner
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] swapTokens #
  - modifiers: lockTheSwap
- [Prv] buyBackTokens #
  - modifiers: lockTheSwap
- [Prv] swapTokensForEth #
- [Prv] swapETHForTokens #
- [Prv] addLiquidity #
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _transferBothExcluded #
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [Prv] calculateTaxFee
- [Prv] calculateLiquidityFee
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Pub] isExcludedFromFee
- [Pub] excludeFromFee #

- modifiers: onlyOwner
- [Pub] includeInFee #
  - modifiers: onlyOwner
- [Ext] setTaxFeePercent #
  - modifiers: onlyOwner
- [Ext] setTaxes #
  - modifiers: onlyOwner
- [Ext] setHappyHourTaxes #
  - modifiers: onlyOwner
- [Ext] startHappyHour #
  - modifiers: onlyOwner
- [Ext] endHappyHour #
  - modifiers: onlyOwner
- [Ext] setMaxTxAmount #
  - modifiers: onlyOwner
- [Ext] setMaxWalletAmount #
  - modifiers: onlyOwner
- [Ext] setMarketingDivisor #
  - modifiers: onlyOwner
- [Ext] setNumTokensSellToAddToLiquidity #
  - modifiers: onlyOwner
- [Ext] setMarketingAddress #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Ext] prepareForPreSale #
  - modifiers: onlyOwner
- [Ext] afterPreSale #
  - modifiers: onlyOwner
- [Prv] transferToAddressETH #
- [Ext] <Fallback> ($)

($) = 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.

Notes:

- addLiquidity function is not used.
- buyBackTokens function is not used.
- _SliquidityFee stays unused.
- swapTokens swaps only tLiquidity amount, remain contract balance may stay unused.

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

- Owner can change tax and liquidity fees.

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}
```

- Owner can change maximum transaction amount.

```
ftrace | funcSig
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner() {
    _maxTxAmount = maxTxAmount;
}
```

- Owner can start and end happy hour (reduced taxes).

```
function startHappyHour() external onlyOwner() {
    _BpreviousLiquidityFee=_BliquidityFee;
    _SpreviousLiquidityFee=_SliquidityFee;
    _prevMarketingFee=_MarketingFee;
    _prevCharityFee=_CharityFee;
    _BliquidityFee=_HappyBuyliquidityFee;
    _SliquidityFee=_HappySellliquidityFee;
    _MarketingFee=_HappyMarketingFee;
    _CharityFee=_HappyCharityFee;
}

    ftrace | funcSig
    function endHappyHour() external onlyOwner() {
    _BliquidityFee=_BpreviousLiquidityFee;
    _SliquidityFee=_SpreviousLiquidityFee;
    _MarketingFee=_prevMarketingFee;
    _CharityFee=_prevCharityFee;
}
```

- Owner can change happy taxes.

```
    function setHappyHourTaxes(uint256 buyLiq⬆,uint256 sellLiq⬆, uint256 marketing⬆,uint256 charity⬆) external onlyOwner() {
    _HappyBuyliquidityFee=buyLiq⬆;
    _HappySellliquidityFee=sellLiq⬆;
    _HappyMarketingFee=marketing⬆;
    _HappyCharityFee=charity⬆;
}
```

- Owner can change taxes.

```
function setTaxes(uint256 buyLiq⬆,uint256 sellLiq⬆, uint256 marketing⬆,uint256 charity⬆) external onlyOwner() {
    _BliquidityFee=buyLiq⬆;
    _SliquidityFee=sellLiq⬆;
    _MarketingFee=marketing⬆;
    _CharityFee=charity⬆;
}
```

- Owner can exclude from the fee.

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

- Owner can change marketingDivisor.

```
ftrace | funcSig
function setMarketingDivisor(uint256 divisor⬆) external onlyOwner() {
    marketingDivisor = divisor⬆;
}
```

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

```
ftrace | funcSig
function setNumTokensSellToAddToLiquidity(uint256 _minimumTokensBeforeSwap⬆) external onlyOwner() {
    minimumTokensBeforeSwap = _minimumTokensBeforeSwap⬆;
}
```

- Owner can change _maxWalletAmount.

```
    function setMaxWalletAmount(uint256 maxTxAmount) external onlyOwner() {
    _maxWalletAmount = maxTxAmount;
}
```

- Owner can change marketing address.

```
ftrace | funcSig
function setMarketingAddress(address _marketingAddress) external onlyOwner() {
    marketingAddress = payable(_marketingAddress);
}
```

- Owner can enable before and after presale modes.

```
ftrace | funcSig
function prepareForPreSale() external onlyOwner {
    setSwapAndLiquifyEnabled(false);
    _taxFee = 0;
    _BliquidityFee = 0;
    _SliquidityFee = 0;
    _MarketingFee = 0;
    _CharityFee = 0;
    _maxTxAmount = 1000000000 * 10**9;
    _maxWalletAmount = 1000000000 * 10**9;
}

ftrace | funcSig
function afterPreSale() external onlyOwner {
    setSwapAndLiquifyEnabled(true);
    _taxFee = 0;
    _BliquidityFee = 5;
    _SliquidityFee = 4;
    _MarketingFee = 3;
    _CharityFee = 3;
    _maxTxAmount = 20000000 * 10**9;
    _maxWalletAmount = 20000000 * 10**9;
}
```

- Owner can lock and unlock. By the way, using these functions the owner could retake privileges even after the ownership was renounced.

```
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = block.timestamp + time;
    emit OwnershipTransferred(_owner, address(0));
}

function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(block.timestamp > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

Conclusion

Smart contracts do not contain high severity issues! Liquidity pair contract's security is not checked due to out of scope. All of the liquidity goes to marketing and charity addresses. The further transfers and operations with the funds raise are not related to this particular contract.

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.