



**TechRate**  
AUDIT COMPANY

# Smart Contract Security Audit

TechRate

June, 2021

# Audit Details



Audited project

**Viola**



Deployer address

**0xA6b384F9201646A95E7a69A4909548502444074E**



Client contacts:

**Viola 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 Viola to perform an audit of smart contracts:

- <https://bscscan.com/address/0x8Acb84E58F24C127a51069D9Eef1c9a8d8F42C70#code>
- <https://bscscan.com/address/0xB543F2D4dBC821291957081b7A879417DeD1cF04#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 23.06.2021

---

Contract name	Viola
Contract address	0x8Acb84E58F24C127a51069D9Eef1c9a8d8F42C70
Total supply	31,247.048716
Token ticker	VIOLA
Decimals	18
Token holders	588
Transactions count	33,064
Top 100 holders dominance	99.80%
Contract deployer address	0xA6b384F9201646A95E7a69A4909548502444074E
Contract's current owner address	0xb543f2d4dbc821291957081b7a879417ded1cf04

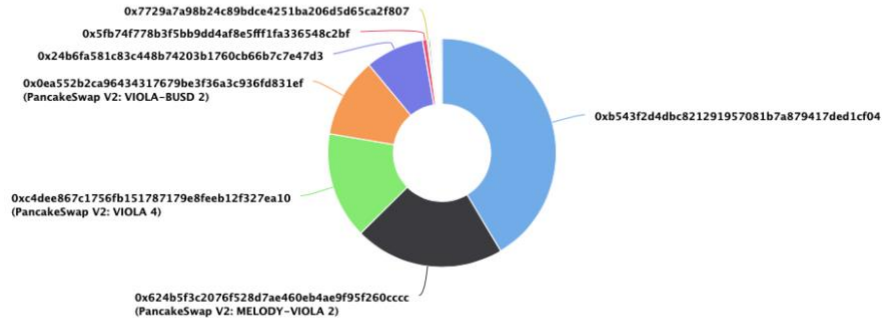
# Viola Token Distribution

The top 100 holders collectively own 99.80% (31,185.62 Tokens) of Viola Token

Token Total Supply: 31,247.05 Token | Total Token Holders: 587

Viola Token Top 100 Token Holders

Source: BscScan.com



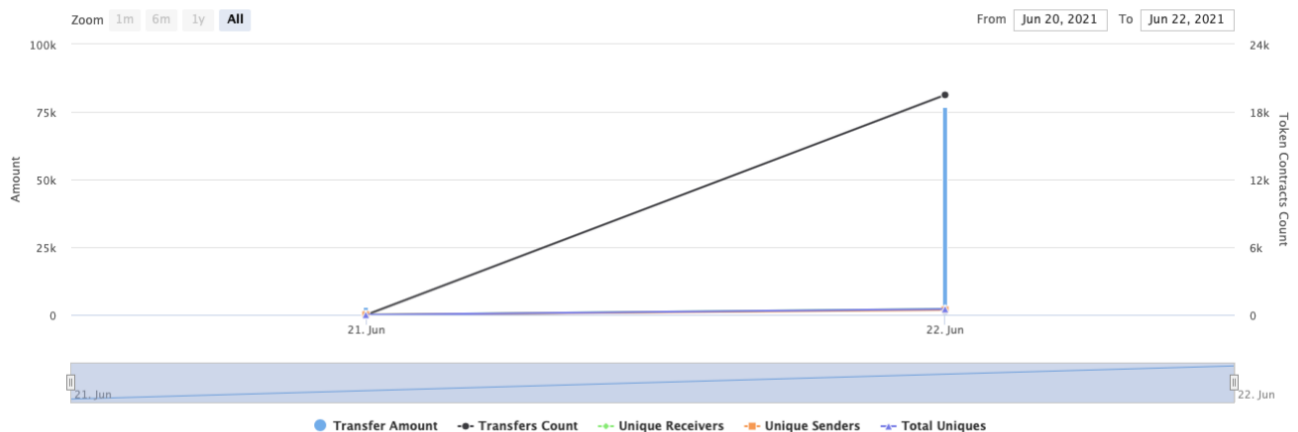
(A total of 31,185.62 tokens held by the top 100 accounts from the total supply of 31,247.05 token)

# Viola Contract Interaction Details

Time Series: Token Contract Overview






Mon 21, Jun 2021 - Tue 22, Jun 2021

Token Contract 0x8AcB84E58F24C127a51069D9Eef1c9a8d8F42C70 (Viola Token)  
Source: BscScan.com





# Viola Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	 0xb543f2d4dbc821291957081b7a879417ded1cf04	12,945.641819608018423658	41.4300%
2	 PancakeSwap V2: MELODY-VIOLA 2	6,616.841026210881314314	21.1759%
3	 PancakeSwap V2: VIOLA 4	4,712.63062150707412591	15.0818%
4	 PancakeSwap V2: VIOLA-BUSD 2	3,525.696760602777885546	11.2833%
5	 0x24b6fa581c83c448b74203b1760cb66b7c7e47d3	2,593.441874254837511095	8.2998%
6	0x5fb74f778b3f5bb9dd4af8e5ff1fa336548c2bf	197.970779149157657371	0.6336%
7	0x7729a7a98b24c89bdce4251ba206d5d65ca2f807	71.366507207062853615	0.2284%
8	0x2fdb469b4fb8c8611dca53e956feebcdd6121b7a	69.601006386051487048	0.2227%
9	0x25bd8058bc833af99915547264e719f4456d696a	42.780601468963004616	0.1369%
10	0xa0f3385af0aed310a545c8fc221c8d5c7a760e86	37.794878723249116968	0.1210%



# MasterChef functions details

## + [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

## + [Int] IBEP20

- [Ext] totalSupply
- [Ext] decimals
- [Ext] symbol
- [Ext] name
- [Ext] getOwner
- [Ext] balanceOf
- [Ext] transfer #
- [Ext] allowance
- [Ext] approve #
- [Ext] transferFrom #

## + [Lib] Address

- [Int] isContract
- [Int] sendValue #
- [Int] functionCall #
- [Int] functionCall #
- [Int] functionCallWithValue #
- [Int] functionCallWithValue #
- [Int] functionStaticCall
- [Int] functionStaticCall
- [Int] functionDelegateCall #
- [Int] functionDelegateCall #
- [Prv] \_verifyCallResult

## + [Lib] SafeBEP20

- [Int] safeTransfer #
- [Int] safeTransferFrom #
- [Int] safeApprove #
- [Int] safeIncreaseAllowance #
- [Int] safeDecreaseAllowance #
- [Prv] \_callOptionalReturn #

## + Context

- [Int] \_msgSender
- [Int] \_msgData



- + Ownable (Context)
  - [Int] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner
  
- + ReentrancyGuard
  - [Int] <Constructor> #
  
- + BEP20 (Context, IBEP20, Ownable)
  - [Pub] <Constructor> #
  - [Ext] getOwner
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] mint #
    - modifiers: onlyOwner
  - [Int] \_transfer #
  - [Int] \_mint #
  - [Int] \_burn #
  - [Int] \_approve #
  - [Int] \_burnFrom #
  
- + ViolaToken (BEP20)
  - [Pub] <Constructor> #
    - modifiers: BEP20
  - [Pub] mint #
    - modifiers: onlyOwner
  - [Int] \_transfer #
    - modifiers: antiWhale
  - [Pub] transferFrom #
  - [Ext] delegates
  - [Ext] delegate #
  - [Ext] delegateBySig #
  - [Ext] getCurrentVotes
  - [Ext] getPriorVotes
  - [Int] \_delegate #
  - [Int] \_moveDelegates #
  - [Int] \_writeCheckpoint #
  - [Int] safe32
  - [Pub] maxTransferAmount
  - [Pub] isExcludedFromAntiWhale
  - [Pub] excludeFromAntiWhale #
    - modifiers: onlyOperator
  - [Pub] updateBurnRate #

- modifiers: onlyOperator
- [Pub] updateMaxTransferAmountRate #
  - modifiers: onlyOperator
- [Int] getChainId
- [Prv] isContract
- + MasterChef (Ownable, ReentrancyGuard)
  - [Pub] <Constructor> #
  - [Ext] poolLength
  - [Pub] add #
    - modifiers: onlyOwner, nonDuplicated
  - [Pub] set #
    - modifiers: onlyOwner
  - [Pub] getMultiplier
  - [Ext] pendingViola
  - [Pub] canHarvest
  - [Pub] massUpdatePools #
  - [Pub] updatePool #
  - [Pub] deposit #
    - modifiers: nonReentrant
  - [Pub] withdraw #
    - modifiers: nonReentrant
  - [Pub] emergencyWithdraw #
    - modifiers: nonReentrant
  - [Int] payOrLockupPendingViola #
  - [Int] safeViolaTransfer #
  - [Pub] dev #
  - [Pub] setFeeAddress #
  - [Pub] updateEmissionRate #
    - modifiers: onlyOwner

(\$) = 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.	Medium issues
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

### 1. Wrong transfer amount

Issue:

- In functions `_transfer(address sender, ...)` and `transferFrom(address sender, ...)` parent transfer invokes with including burning amount. This is not acceptable, because burn part of this amount is already burned and not exist any more.

Recommendation:

Subtract `burnAmount` from amount before parent transfer function call.

**Viola team comments:**

if we send 100, my balance will be decreased by 102 (2 is burn amount) and receipts balance will be increased by 100. so it should be `amount - burnAmount`.

## ✓ Low Severity Issues

### 1. Block gas limit

Issue:

`add(uint256 _allocPoint, ...)`, `set(uint256 _pid, ...)` and `updateEmissionRate()` could invoke `massUpdatePools()` function, that can fail due to block gas limit if the pool size is too big.

## Owner privileges

- Owner can change the operator of the referral contract.
- Operator can exclude from anti whale.
- Operator can change burn rate.
- Operator can change the max transfer amount rate.

# Conclusion

Smart contracts contain medium severity issues.

---

## *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.*



[Techrate1](#)



[Techrate](#)



[Techrate audits](#)