TechRate
August, 2022

TECH
RATE

SMART CONTRACTS SECURITY AUDIT REPORT

Techrate_audits Techrate Techrate1

Audit Details

Audited project

VRFI

Deployer address

0x2fC7d6874731828a503f40D57BE7c71b544BFD7A

Client contacts:

VRFI team

Blockchain

Ethereum

Project website:

http://verifidefi.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 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 VRFI to perform an audit of smart contracts:

https://etherscan.io/address/0x497d563072C89cf33be4aA692233FC33fC0eE2a9#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.

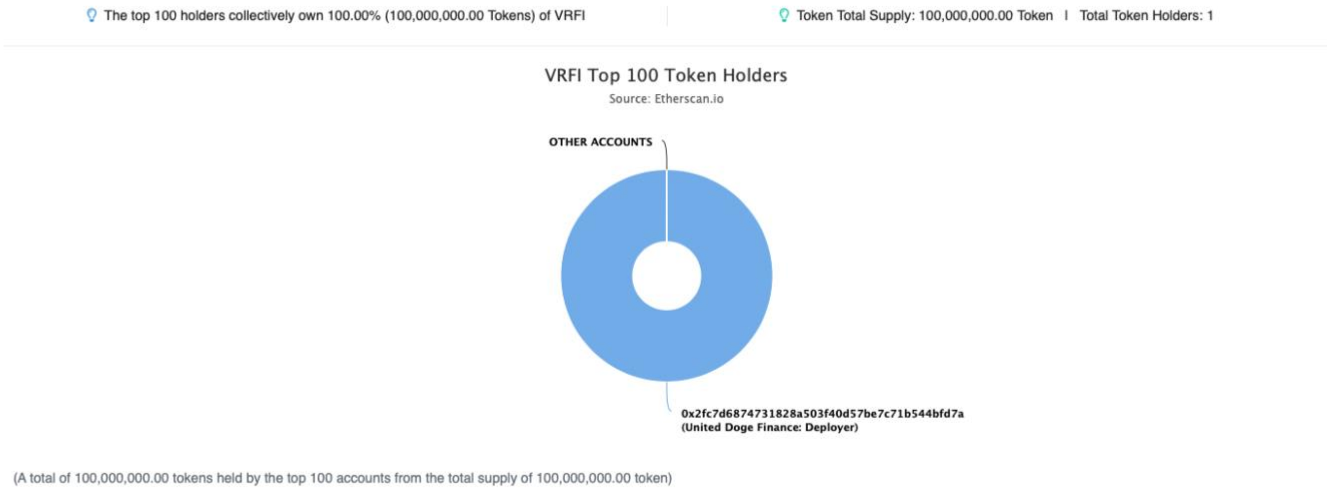
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 11.08.2022

Contract name	VRFI
Contract address	0x497d563072C89cf33be4aA692233FC33fC0eE2a9
Total supply	100,000,000
Token ticker	VRFI
Decimals	18
Token holders	1
Transactions count	1
Top 100 holders dominance	100.00%
Insurance wallet	0x6cf48b5edc8b4e72d2b67c5fa6a46b43358f8181
Deployer wallet	0xca57fcfbc10e6e58ec793b644797d3570dbefc56
Salaries/Dev/Marketing wallet	0x9361eae77a347691e306f11db8d8f9c961aff3cc
Uniswap V2 pair	0x888b11bb6c18b5802e8aaa1de0ffcfbedab8d613
Contract deployer address	0x2fC7d6874731828a503f40D57BE7c71b544BFD7A
Owner address	0x2fc7d6874731828a503f40d57be7c71b544bfd7a

VRFI Token Distribution



The top 100 holders collectively own 100.00% (100,000,000.00 Tokens) of VRFI

Token Total Supply: 100,000,000.00 Token | Total Token Holders: 1

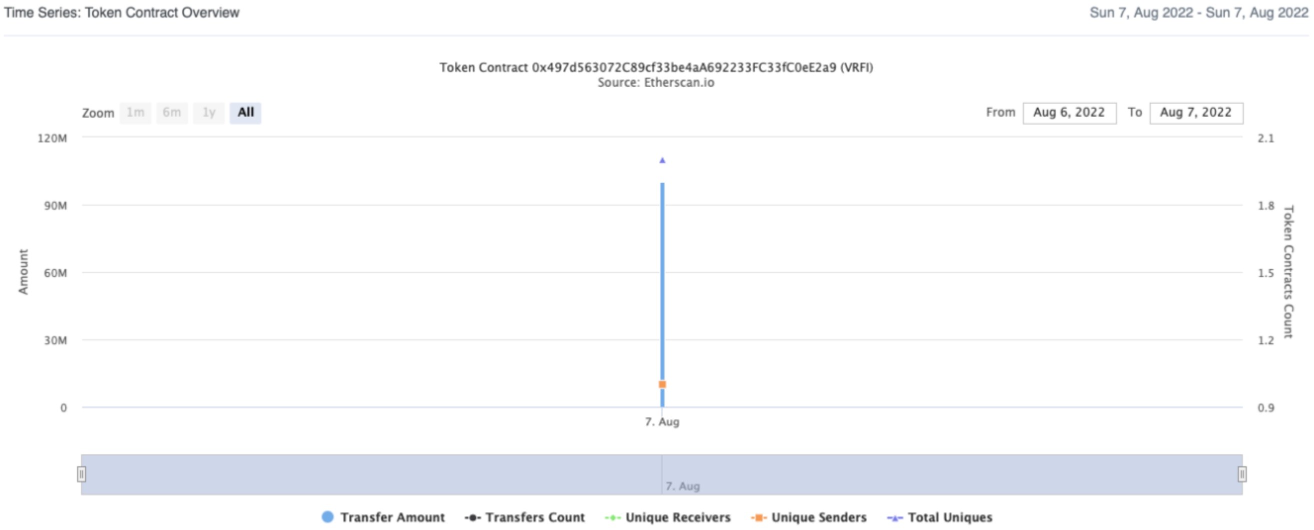
VRFI Top 100 Token Holders

Source: Etherscan.io

OTHER ACCOUNTS
0x2fc7d6874731828a503f40d57be7c71b544bfd7a
(United Doge Finance: Deployer)

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

VRFI Contract Interaction Details



Time Series: Token Contract Overview

Sun 7, Aug 2022 - Sun 7, Aug 2022

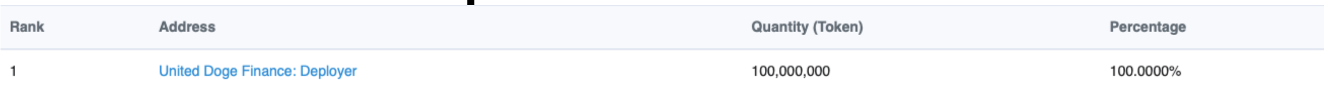
Token Contract 0x497d563072C89cf33be4aA692233FC33fC0eE2a9 (VRFI)

Source: Etherscan.io

Zoom 1m 6m 1y All

From Aug 6, 2022 To Aug 7, 2022

VRFI Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	United Doge Finance: Deployer	100,000,000	100.0000%

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

 + [Int] IERC20Metadata (IERC20)
   - [Ext] name
   - [Ext] symbol
   - [Ext] decimals

 + [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 #

+ [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 #

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

+  ERC20 (Context, IERC20, IERC20Metadata)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Int] _transfer #
  - [Int] _mint #
  - [Int] _burn #
  - [Int] _approve #
  - [Int] _spendAllowance #
  - [Int] _beforeTokenTransfer #
  - [Int] _afterTokenTransfer #

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

+  VRFI (ERC20, Ownable)
  - [Pub] <Constructor> #
```

```
    - modifiers: ERC20
  - [Ext] <Fallback> ($)
  - [Pub] updateUniswapV2Router #
    - modifiers: onlyOwner
  - [Ext] enableTrading #
    - modifiers: onlyOwner
  - [Pub] excludeFromFees #
    - modifiers: onlyOwner
  - [Pub] excludeMultipleAccountsFromFees #
    - modifiers: onlyOwner
  - [Ext] setSwapAtAmount #
    - modifiers: onlyOwner
  - [Ext] setMaxWallet #
    - modifiers: onlyOwner
  - [Ext] setMaxTx #
    - modifiers: onlyOwner
  - [Ext] setDailyLimit #
    - modifiers: onlyOwner
  - [Ext] setWallets #
    - modifiers: onlyOwner
  - [Ext] setSellFee #
    - modifiers: onlyOwner
  - [Ext] setSevenDayFee #
    - modifiers: onlyOwner
  - [Ext] setBuyFee #
    - modifiers: onlyOwner
  - [Ext] setTransferFee #
    - modifiers: onlyOwner
  - [Pub] setAutomatedMarketMakerPair #
    - modifiers: onlyOwner
  - [Prv] _setAutomatedMarketMakerPair #
  - [Ext] claimStuckTokens #
    - modifiers: onlyOwner
  - [Pub] isExcludedFromFees
  - [Int] _transfer #
  - [Prv] swapAndSendToFee #
  - [Prv] swapAndLiquify #
  - [Prv] swapTokensForEth #
  - [Prv] addLiquidity #


($) = 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 excludeMultipleAccountsFromFees() uses the loop to exclude multiple addresses from the fees list. Function will be aborted with OUT_OF_GAS exception if there will be a long addresses list.

Recommendation:

Check that the array length is not too big.

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

- Owner can change uniswapV2Router.
- Owner can enable trading.
- Owner can exclude addresses from fees.
- Owner can change swapTokensAtAmount and timeGap.
- Owner can change maxWalletAmount.
- Owner can change maxTxAmount.
- Owner can change dailyLimit.
- Owner can change fee receivers addresses.
- Owner can change fees.
- Owner can mark addresses as pairs.
- Owner can withdraw contract ERC20 tokens and ETHs.

Testnet deployment

Contracts Description Table

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	transfer	Public ❗	🛑	NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
VRFI	Implementation	ERC20, Ownable		
└	enableTrading	External ❗	🛑	onlyOwner
└	excludeFromFees	Public ❗	🛑	onlyOwner
└	excludeMultipleAccountsFromFees	Public ❗	🛑	onlyOwner
└	setSwapAtAmount	External ❗	🛑	onlyOwner
└	setMaxWallet	External ❗	🛑	onlyOwner
└	setMaxTx	External ❗	🛑	onlyOwner
└	setDailyLimit	External ❗	🛑	onlyOwner
└	setWallets	External ❗	🛑	onlyOwner
└	setSellFee	External ❗	🛑	onlyOwner
└	setSevenDayFee	External ❗	🛑	onlyOwner
└	setBuyFee	External ❗	🛑	onlyOwner
└	setTransferFee	External ❗	🛑	onlyOwner
└	setAutomatedMarketMakerPair	Public ❗	🛑	onlyOwner
└	claimStuckTokens	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope. The further transfers and operations with the funds raise are not related to this particular contract.

Liquidity locking details are 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.