Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

October, 2021

Audit Details

Audited project

ForeverDOGE

Deployer address

0x0F19d97b1dE63230d5B5be3625d57BE0adEA226A

Client contacts:

ForeverDOGE 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 ForeverDOGE to perform an audit of smart contracts:

https://bscscan.com/address/0x6DfFd171150D2d3b128760aa866512BAb3273612#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

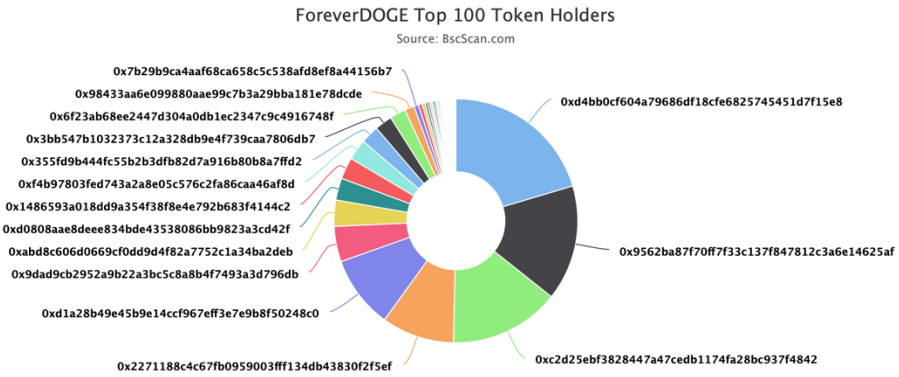
Token contract details for 05.10.2021

Contract name	ForeverDOGE
Contract address	0x6DfFd171150D2d3b128760aa866512BAb3273612
Total supply	2,500,000,000,000
Token ticker	ForeverDOGE
Decimals	9
Token holders	111
Transactions count	1,126
Top 100 holders dominance	10,363.12%
Contract deployer address	0x0F19d97b1dE63230d5B5be3625d57BE0adEA226A
Contract’s current owner address	0x0F19d97b1dE63230d5B5be3625d57BE0adEA226A

ForeverDOGE Token Distribution

The top 100 holders collectively own 10,363.12% (259,078,040,638,557.00 Tokens) of ForeverDOGE

Token Total Supply: 2,500,000,000,000.00 Token | Total Token Holders: 111

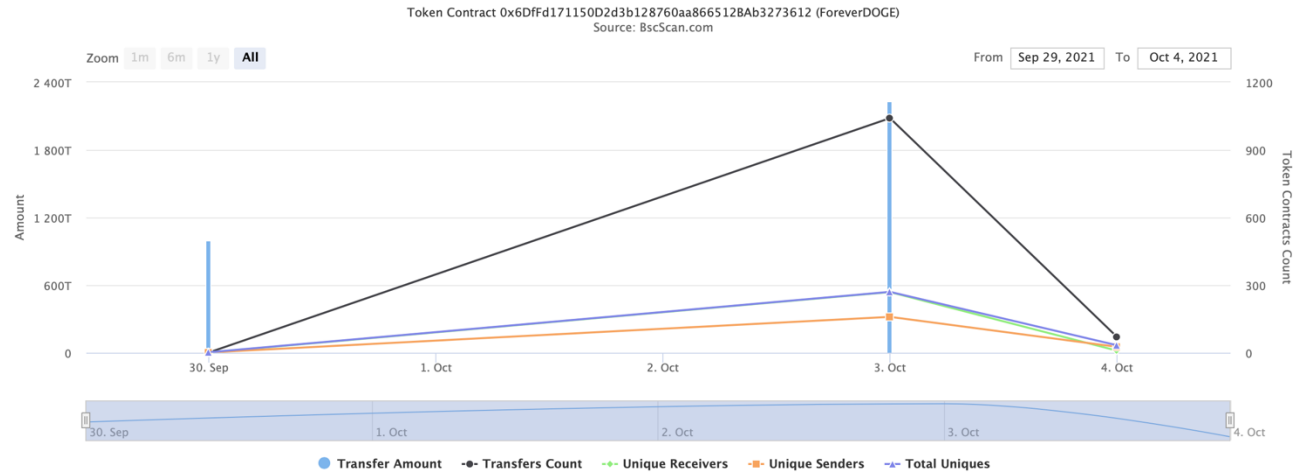


(A total of 259,078,040,638,557.00 tokens held by the top 100 accounts from the total supply of 2,500,000,000,000.00 token)

ForeverDOGE Contract Interaction Details

Time Series: Token Contract Overview

Thu 30, Sept 2021 - Mon 4, Oct 2021



ForeverDOGE Top 10 Token Holders

Rank	Address	Quantity	Percentage
1	0xd4bb0cf604a79686df18cfe6825745451d7f15e8	52,839,166,788,116.823527422	2,113.5667%
2	0x9562ba87f70ff7f33c137f847812c3a6e14625af	39,633,846,450,199.720609451	1,585.3539%
3	0xc2d25ebf3828447a47cedb1174fa28bc937f4842	37,881,213,180,441.022731399	1,515.2485%
4	0x2271188c4c67fb0959003fff134db43830f2f5ef	25,018,367,789,224.335722301	1,000.7347%
5	0xd1a28b49e45b9e14ccf967eff3e7e9b8f50248c0	24,904,697,782,824.008943389	996.1879%
6	0x9dad9cb2952a9b22a3bc5c8a8b4f7493a3d796db	12,134,026,582,800.658419742	485.3611%
7	0xabd8c606d0669cf0dd9d4f82a7752c1a34ba2deb	9,000,000,000,000	360.0000%
8	0xd0808aae8deee834bde43538086bb9823a3cd42f	7,559,999,999,999.997159937	302.4000%
9	0x1486593a018dd9a354f38f8e4e792b683f4144c2	7,395,563,817,267.192	295.8226%
10	0xf4b97803fed743a2a8e05c576c2fa86caa46af8d	7,200,000,000,000.038061573	288.0000%

Contract functions details

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

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

+ [Int] InterfaceLP
  - [Ext] sync #

+ ERC20Detailed (IERC20)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals

+ [Lib] SafeMathInt
  - [Int] mul
  - [Int] div
  - [Int] sub
  - [Int] add
  - [Int] abs

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

+ [Int] IDEXFactory
  - [Ext] createPair #

+ [Int] IDEXRouter
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidity #
  - [Ext] addLiquidityETH ($)

- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
- [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ ForeverDOGE (ERC20Detailed, Ownable)
  - [Ext] rebase #
    - modifiers: onlyMaster
  - [Pub] <Constructor> #
    - modifiers: ERC20Detailed
  - [Ext] setMaster #
    - modifiers: onlyOwner
  - [Ext] setLP #
    - modifiers: onlyOwner
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
    - modifiers: validRecipient,initialDistributionLock
  - [Ext] allowance
  - [Ext] transferFrom #
    - modifiers: validRecipient
  - [Int] _transferFrom #
  - [Int] _basicTransfer #
  - [Int] takeFee #
  - [Int] swapBack #
    - modifiers: swapping
  - [Ext] approve #
    - modifiers: initialDistributionLock
  - [Ext] increaseAllowance #
    - modifiers: initialDistributionLock
  - [Ext] decreaseAllowance #
    - modifiers: initialDistributionLock
  - [Ext] setInitialDistributionFinished #
    - modifiers: onlyOwner
  - [Ext] enableTransfer #
    - modifiers: onlyOwner
  - [Ext] setFeeExempt #
    - modifiers: onlyOwner
  - [Ext] checkFeeExempt
  - [Ext] setMaxWalletExempt #
    - modifiers: onlyOwner
  - [Ext] checkMaxWalletExempt
  - [Ext] setMaxWalletToken #
    - modifiers: onlyOwner
  - [Ext] checkMaxWalletToken
  - [Int] shouldTakeFee
  - [Int] shouldSwapBack
  - [Ext] setSwapBackSettings #
    - modifiers: onlyOwner
  - [Ext] setTargetLiquidity #
    - modifiers: onlyOwner
  - [Ext] isNotInSwap
  - [Ext] checkSwapThreshold
  - [Ext] manualSync #
  - [Ext] setFees #
    - modifiers: onlyOwner

- [Ext] setFeeReceivers #
  - modifiers: onlyOwner
- [Pub] rescueToken #
  - modifiers: onlyOwner
- [Ext] clearStuckBalance #
  - modifiers: onlyOwner
- [Prv] transferToAddressETH #
- [Pub] getCirculatingSupply
- [Ext] sendPresale #
  - modifiers: onlyOwner
- [Pub] getLiquidityBacking
- [Pub] isOverLiquified
- [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.	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.	Low issues
18.	Design Logic.	Medium issues
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. No transfer event emitted

Issue:

- The function `_basicTransfer()` do not emit Transfer event

Recommendation:
Add Transfer event to function.

Low Severity Issues

2. sendPresale issues

Issue:

- The function `sendPresale()` uses the loop distribute values from values list to recipients from recipients list. Function will be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

Recommendation:
Check that the excluded array length is not too big.

- The function `sendPresale()` do not check recipients list length and values list length with each other to avoid mismatch.

Recommendation:
Check arrays' lengths to stabilize function working.

```
function sendPresale(address[] calldata recipients, uint256[] calldata values)
    external
    onlyOwner
{
  for (uint256 i = 0; i < recipients.length; i++) {
    _transferFrom(msg.sender, recipients[i], values[i]);
  }
}
```

3. Rounding errors

Issue:

- At some calculation with division, it is goes first. In Solidity we don't have floating points, but instead we get rounding errors.

Recommendation:

Do division after multiplication.

```
uint256 public gonMaxWallet = TOTAL_GONS.div(100).mul(5);
function setMaxWalletToken(uint256 _num, uint256 _denom)
    external
    onlyOwner
{
    gonMaxWallet = TOTAL_GONS.div(_denom).mul(_num);
}
function setSwapBackSettings(
    bool _enabled,
    uint256 _num,
    uint256 _denom
) external onlyOwner {
    swapEnabled = _enabled;
    gonSwapThreshold = TOTAL_GONS.div(_denom).mul(_num);
}
```

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

- Owner can change master of the contract.

```
function setMaster(address _master) external onlyOwner {
    master = _master;
}
```

- Owner can change contract pair.

```
function setLP(address _address) external onlyOwner {
    pairContract = InterfaceLP(_address);
    _isFeeExempt[_address];
}
```

- Owner can finish initial distribution (allows transfers).

```
function setInitialDistributionFinished() external onlyOwner {
    initialDistributionFinished = true;
}
```

- Owner can exclude from transfer, maxWallet and fee restrictions.

```
function enableTransfer(address _addr) external onlyOwner {
    allowTransfer[_addr] = true;
}

function setFeeExempt(address _addr) external onlyOwner {
    _isFeeExempt[_addr] = true;
}

function checkFeeExempt(address _addr) external view returns (bool) {
    return _isFeeExempt[_addr];
}

function setMaxWalletExempt(address _addr) external onlyOwner {
    _isMaxWalletExempt[_addr] = true;
}
```

- Owner can change maxWallet amount.

```
function setMaxWalletToken(uint256 _num, uint256 _denom)
    external
    onlyOwner
{
    gonMaxWallet = TOTAL_GONS.div(_denom).mul(_num);
}
```

- Owner can change swapBack settings.

```
function setSwapBackSettings(
    bool _enabled,
    uint256 _num,
    uint256 _denom
) external onlyOwner {
    swapEnabled = _enabled;
    gonSwapThreshold = TOTAL_GONS.div(_denom).mul(_num);
}
```

- Owner can change target liquidity.

```
function setTargetLiquidity(uint256 target, uint256 accuracy) external onlyOwner {
    targetLiquidity = target;
    targetLiquidityDenominator = accuracy;
}
```

- Owner can change fees and fee receivers addresses.

```
function setFees(
    uint256 _ecosystemFee,
    uint256 _liquidityFee,
    uint256 _buyBackFee,
    uint256 _marketingFee,
    uint256 _feeDenominator
) external onlyOwner {
    ecosystemFee = _ecosystemFee;
    liquidityFee = _liquidityFee;
    buyBackFee = _buyBackFee;
    marketingFee = _marketingFee;
    totalFee = ecosystemFee.add(liquidityFee).add(marketingFee).add(buyBackFee);
    feeDenominator = _feeDenominator;
    require(totalFee < feeDenominator / 4);
}

function setFeeReceivers(
    address _autoLiquidityReceiver,
    address _ecosystemFeeReceiver,
    address _marketingFeeReceiver,
    address _buyBackFeeReceiver
) external onlyOwner {
    autoLiquidityReceiver = _autoLiquidityReceiver;
    ecosystemFeeReceiver = _ecosystemFeeReceiver;
    marketingFeeReceiver = _marketingFeeReceiver;
    buyBackFeeReceiver = _buyBackFeeReceiver;
}
```

- **Owner can withdraw contract tokens and BNBs.**

```
function rescueToken(address tokenAddress, uint256 tokens)
    public
    onlyOwner
    returns (bool success)
{
    return ERC20Detailed(tokenAddress).transfer(msg.sender, tokens);
}

function clearStuckBalance(uint256 amountPercentage, address adr) external onlyOwner {
    uint256 amountETH = address(this).balance;
    payable(adr).transfer(
        (amountETH * amountPercentage) / 100
    );
}
```

Conclusion

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

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.