Smart Contract Security Audit

Audit details:

Audited project:	Ape Token
Deployer address:	0x738CDD9461d0a218287123b2384d38610A447e21
Client contacts:	Ape Token team
Blockchain:	Binance Smart Chain
Project website:	www.apetoken.net

May, 2021
TechRate

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 Ape Token to perform an audit of smart contracts:

- *https://bscscan.com/address/0x196dc2421af765c2673fc6093fb1374b28064270*

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

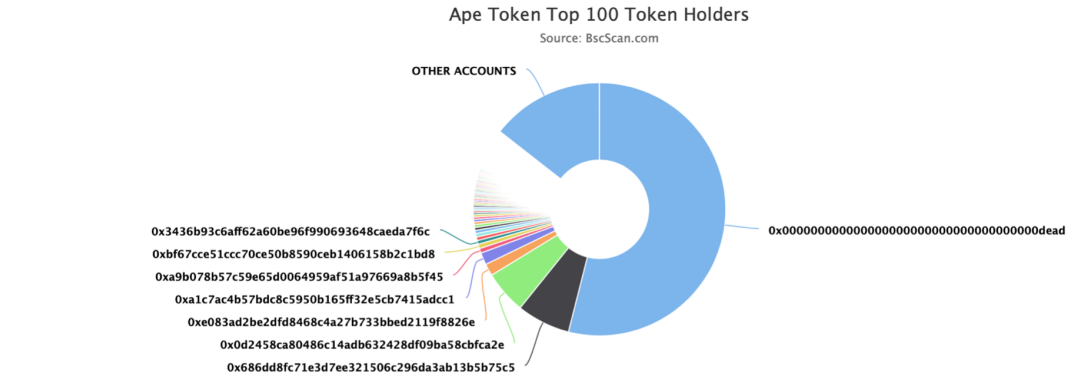
Token contract details for 19.05.2021.

Contract name:	Ape Token
Contract address:	0x196dC2421AF765c2673FC6093fb1374b28064270
Total supply:	1_000_000_000_000_000_000_000_000
Token ticker:	APET
Decimals:	9
Token holders:	4,421
Transactions count:	11,112
Top 100 holders dominance:	85.55%
Liquidity fee:	5
Tax fee:	4
Total fees:	79918675482629613545938
Uniswap V2 pair:	0x686dd8fc71e3d7ee321506c296da3ab13b5b75c5
Contract deployer address:	0x738CDD9461d0a218287123b2384d38610A447e21
Contract's current owner address:	0x2b35cd4f8f691af07951cfa1650fc556a56d4828

Ape Token token distribution

The top 100 holders collectively own 85.55% (855,467,162,382,440.00 Tokens) of Ape Token

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 4,421

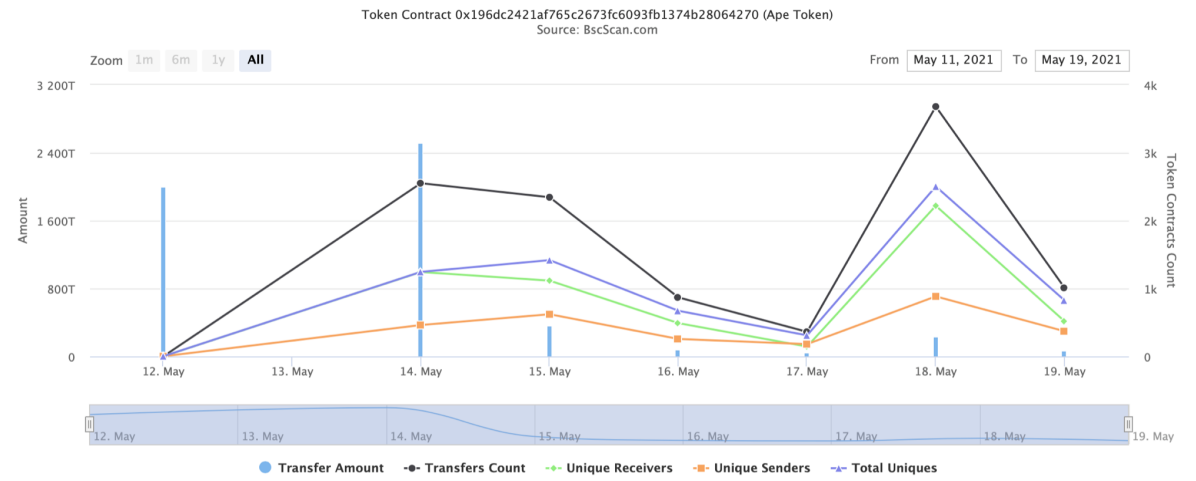


(A total of 855,467,162,382,440.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

Ape Token contract interaction details

Time Series: Token Contract Overview

Wed 12, May 2021 - Wed 19, May 2021



Ape Token top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x000000000000000000000000000000000000dead	539,597,937,027,897.705183872	53.9598%
2	0x686dd8fc71e3d7ee321506c296da3ab13b5b75c5	68,424,036,806,014.551951047	6.8424%
3	0x0d2458ca80486c14adb632428df09ba58cbfca2e	55,004,559,395,337.896307952	5.5005%
4	0xe083ad2be2dfd8468c4a27b733bbed2119f8826e	15,445,840,018,830.167359973	1.5446%
5	0xa1c7ac4b57bdc8c5950b165ff32e5cb7415adcc1	15,405,283,785,232.121521749	1.5405%
6	0xa9b078b57c59e65d0064959af51a97669a8b5f45	5,855,426,256,052.928943088	0.5855%
7	0xbf67cce51ccc70ce50b8590ceb1406158b2c1bd8	5,775,560,970,220.742502069	0.5776%
8	0x3436b93c6aff62a60be96f990693648caeda7f6c	4,725,750,498,898.218504142	0.4726%
9	0x45177d3fe65a203718fc2b0939b80c0ca36ce9d8	4,568,322,627,885.814623662	0.4568%
10	0xf80a9fa8a946942afebf2c96f7bad828aa25771f	4,444,682,530,359.481098429	0.4445%

Ape Token LP token holders

Rank	Address	Quantity	Percentage	Analytics
1	0x8076dcd26c1c49145a9c91cb0e42b818ecba6942	3,433.400493872887402748	54.8506%	
2	0x2d045410f002a95efcee67759a92518fa3fce677	1,595.98426747062400074	25.4968%	
3	0x2b35cd4f8f691af07951cfa1650fc556a56d4828	1,147.412766829098530747	18.3306%	
4	0x07d80ae6f36a5e08dca74ce884a24d39db9934ed	68.454258011429648123	1.0936%	
5	0x1d2daeb7925cc6fc72904d073b63eecec3d292be	13.60831698141230308	0.2174%	
6	0x1e53c076e50c7b271c2d6e9e20388f0d2e461263	0.691863096064424788	0.0111%	
7	0xee02a3eff25e9ae26e5dc6aa52ae5ded271c32db	0.000017046008481559	0.0000%	
8	0x0000000000000000000000000000000000000000	0.000000000000001	0.0000%	

Contract functions details

+ [Int] IERC20
  - [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

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

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

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

- [Pub] transferOwnership #
  - modifiers: onlyOwner
- [Pub] geUnlockTime
- [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] mint #
- [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 #

+ APET (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] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward #
  - modifiers: onlyOwner
- [Ext] includeInReward #
  - modifiers: onlyOwner

- [Pub] excludeFromFee #
  - modifiers: onlyOwner
- [Pub] includeInFee #
  - modifiers: onlyOwner
- [Ext] setTaxFeePercent #
  - modifiers: onlyOwner
- [Ext] setMarketingFeePercent #
  - modifiers: onlyOwner
- [Ext] setLiquidityFeePercent #
  - modifiers: onlyOwner
- [Ext] setMaxTxPercent #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Ext] <Fallback> ($)
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [Prv] _takeMarketing #
- [Prv] calculateTaxFee
- [Prv] calculateMarketingFee
- [Prv] calculateLiquidityFee
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Pub] isExcludedFromFee
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] swapAndLiquify #
  - modifiers: lockTheSwap
- [Prv] swapTokensForEth #
- [Prv] addLiquidity #
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _transferBothExcluded #

($) = 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 `includeInReward()` uses the loop to find and remove addresses from the `_excluded` list. Function will be aborted with `OUT_OF_GAS` exception if there will be a long excluded addresses list.

```
function includeInReward(address account⬆) external onlyOwner() {
    require(_isExcluded[account⬆], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account⬆) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account⬆] = 0;
            _isExcluded[account⬆] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The function `_getCurrentSupply` also uses the loop for evaluating total supply. It also could be aborted with `OUT_OF_GAS` exception if there will be a long excluded addresses list.

```
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
        ) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

Recommendation:

Use EnumerableSet instead of array or do not use long arrays.

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

- ❏ Owner can change the tax, liquidity and marketing fee.

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

function setMarketingFeePercent(uint256 marketingFee) external onlyOwner() {
    _marketingFee = marketingFee;
}

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

- ❏ Owner can change the maximum transaction amount.

```
  function setMaxTxPercent(uint256 maxTxPercent⬆) external onlyOwner() {
      _maxTxAmount = _tTotal.mul(maxTxPercent⬆).div(
          10**2
      );
  }
```

- ❏ Owner can exclude from the fee.

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

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

```
//Locks the contract for owner for the amount of time provided
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = block.timestamp + time;
    emit OwnershipTransferred(_owner, address(0));
}

//Unlocks the contract for owner when _lockTime is exceeds
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 contain low severity issues. LP pair contract is not checked.

Liquidity locking details provided by the team -
https://dxsale.app/app/pages/dxlockview?id=0&add=0x2b35cd4F8f691af07951CFa1650Fc556a56D4828&type=tokenlock&chain=BSC

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.