Smart Contract Security Audit

Audit details:

Audited project:	Secured Marsupial
Deployer address:	0xf103d2aba493749a402b7de11cf31f5844062b74
Client contacts:	Secured Marsupial team
Blockchain:	Binance Smart Chain
Project website:	https://www.marsupial.finance

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 Secured Marsupial to perform an audit of smart contracts:

- *https://bscscan.com/address/0x2aada8138787eb7a8907232da6756c6ac6186ef2#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

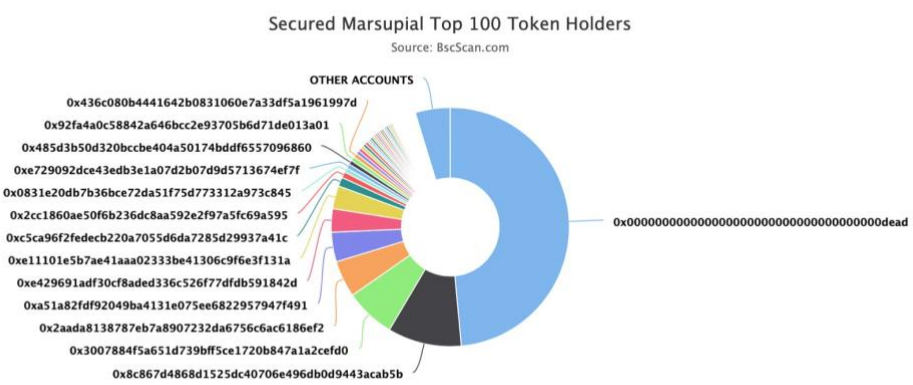
Token contract details for 10.05.2021.

Contract name:	Secured Marsupial
Contract address:	0x2aada8138787eb7a8907232da6756c6ac6186ef2
Total supply:	100000000000000000000000
Token ticker:	SMARSUP
Decimals:	9
Token holders:	1515
Transactions count:	4790
Top 100 holders dominance:	55.24 %
Liquidity fee:	4
Tax fee:	4
Total fees:	52684216788401723403228
Uniswap V2 pair:	0x0831e20db7b36bce72da51f75d773312a973c845
Contract deployer address:	0xf103d2aba493749a402b7de11cf31f5844062b74
Contract’s current owner address:	0x8c867d4868d1525dc40706e496db0d9443acab5b

Secured Marsupial token distribution

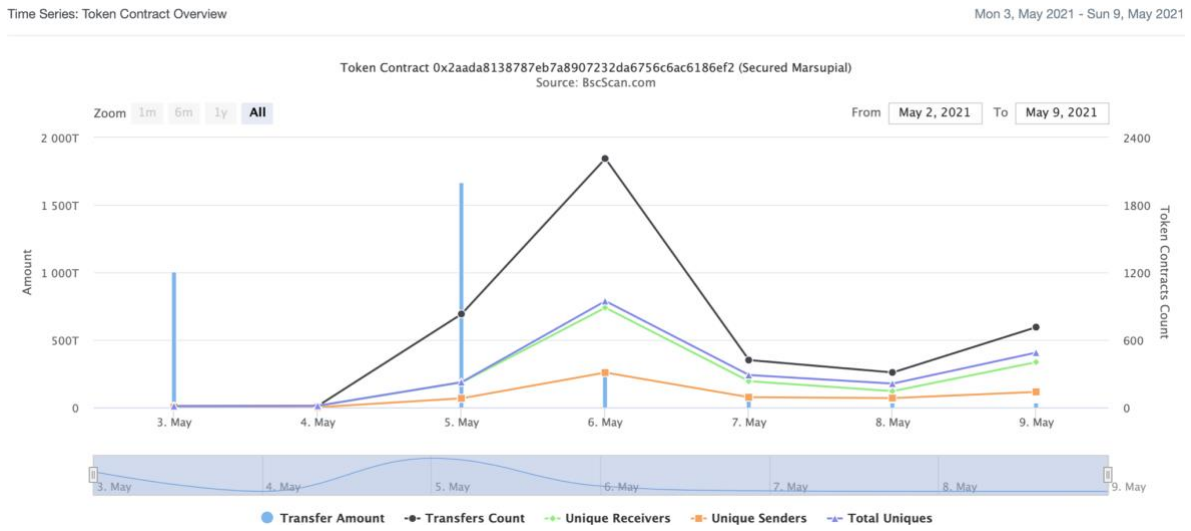
The top 100 holders collectively own 95.24%
(952,431,042,464,364.00 Tokens) of Secured Marsupial

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 1,515



(A total of 952,431,042,464,364.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

Secured Marsupial contract interaction details



Secured Marsupial top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x000000000000000000000000000000000000dead	485,206,777,512,473.866160661	48.5207%
2	0x8c867d4868d1525dc40706e496db0d9443acab5b	100,304,907,080,873.365019183	10.0305%
3	0x3007884f5a651d739bff5ce1720b847a1a2cefd0	67,926,313,805,272.945026475	6.7926%
4	0x2aada8138787eb7a8907232da6756c6ac6186ef2	49,596,594,685,106.357905996	4.9597%
5	0xa51a82fdf92049ba4131e075ee68229579 47f491	40,067,415,493,115.212324691	4.0067%
6	0xe429691adf30cf8aded336c526f77dfdb591842d	31,560,187,809,915.018889246	3.1560%
7	0xe11101e5b7ae41aaa02333be41306c9f6e3f131a	31,227,368,048,247.774068487	3.1227%
8	0xc5ca96f2fedecb220a7055d6da7285d29937a41c	12,347,268,866,743.240952655	1.2347%
9	0x2cc1860ae50f6b236dc8aa592e2f97a5fc69a595	7,994,997,899,382.238727702	0.7995%
10	0x0831e20db7b36bce72da51f75d773312a973c845	6,102,217,853,746.943116684	0.6102%

Secured Marsupial LP token holders

Rank	Address	Quantity	Percentage
1	0x8c867d4868d1525dc40706e496db0d9443acab5b	516.914538391018752298	57.2468%
2	0xeb3a9c56d963b971d320f889be2fb8b59853e449	366.6060555964672	40.6006%
3	0x07d80ae6f36a5e08dca74ce884a24d39db9934ed	18.774859053932618746	2.0793%
4	0x61754dd019c55bb9853e4ed673527129807578bf	0.662247507093454997	0.0733%
5	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] add
  - [Int] sub
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] div
  - [Int] mod
  - [Int] mod

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

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

+ Ownable (Context)
  - [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 #

  + CoinToken (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
    - [Prv] _transferBothExcluded #
    - [Pub] excludeFromFee #
      - modifiers: onlyOwner
    - [Pub] includeInFee #
-     modifiers: onlyOwner
-   - [Ext] setTaxFeePercent #
-     modifiers: onlyOwner
```

- [Ext] setLiquidityFeePercent #
- modifiers: onlyOwner
- [Pub] setNumTokensSellToAddToLiquidity #
- modifiers: onlyOwner
- [Pub] setMaxTxPercent #
- modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
- modifiers: onlyOwner
- [Ext] <Fallback> ($)
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [Pub] claimTokens #
  - modifiers: onlyOwner
- [Prv] calculateTaxFee
- [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 #

($) = 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

- ❏ Owner can change the tax and liquidity fee.
- ❏ Owner can change the maximum transaction amount.
- ❏ Owner can exclude from the fee.
- ❏ Owner can withdraw the funds from the token contract using function claimTokens.

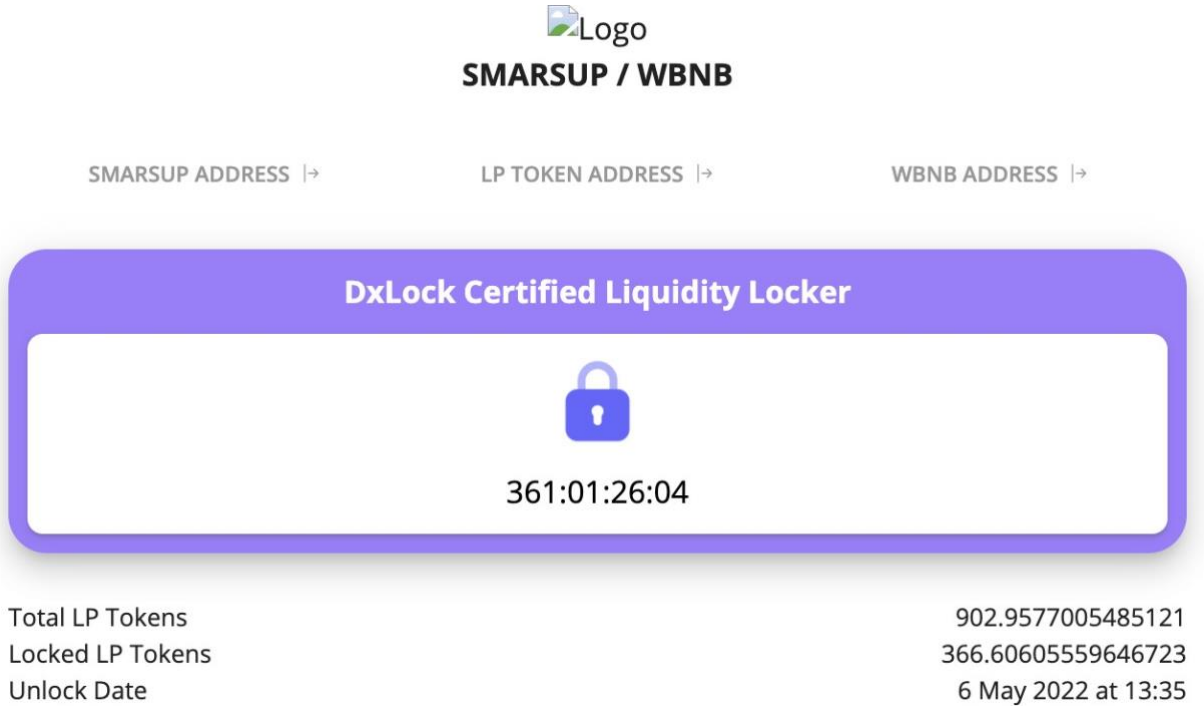
```
function claimTokens() public onlyOwner {
        payable(_owner).transfer(address(this).balance);
}
```

Conclusion

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

Part of the liquidity locking details provided by the team can be found by this link - https://dxsale.app/app/pages/dxlockview?id=0&add=0x8C867d4868d1525dC40706e496dB0D9443ACaB5B&type=lplock&chain=BSC

12-05-2021 Stats: 0% liquidity is unlocked.



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.