TechRate

Blockchain solutions and consulting

Smart Contract Security Audit

Audit details:

Audited project: Safelcarus

Deployer address: 0x58322be0B0679671FBFC0D2B63BbA631d09e07C4

Client contacts: SafeIcarus team

Blockchain: Binance Smart Chain

Project website: https://safeicarus.io/

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 SafeIcarus to perform an audit of smart contracts:

- *https://bscscan.com/address/0xd2f9b51c6a1b5598f0ea51ef95d70cb045692d0f#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 25.05.2021.

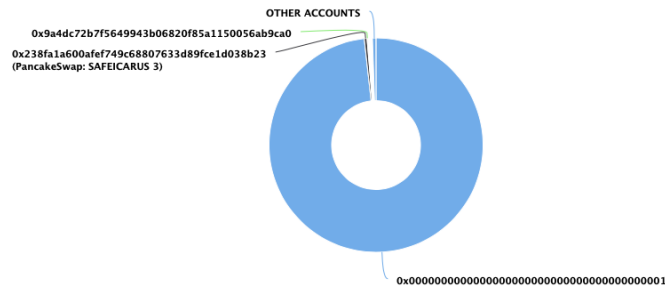|	
Contract name:	SafeIcarus
Contract address:	0xD2f9B51C6a1b5598f0ea51eF95d70cB045692D0F
Total supply:	100000000000000000
Token ticker:	SAFEICARUS
Decimals:	9
Token holders:	44,730
Transactions count:	115,120
Top 100 holders dominance:	99.47%
Liquidity fee:	8
Tax fee:	5
Total fees:	126934717064261406949735
Uniswap V2 pair:	0x238fa1a600afef749c68807633d89fce1d038b23
Contract deployer address:	0x58322be0B0679671FBFC0D2B63BbA631d09e07C4
Contract's current owner address:	0x0000000000000000000000000000000000000000

SafeIcarus token distribution

SafeIcarus Top 100 Token Holders
Source: BscScan.com

OTHER ACCOUNTS

0x9a4dc72b7f5649943b06820f85a1150056ab9ca0

0x238fa1a600afef749c68807633d89fce1d038b23
(PancakeSwap: SAFEICARUS 3)

0x0000000000000000000000000000000000000001

(A total of 99,472,276,350,806,800.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

SafeIcarus contract interaction details

Time Series: Token Contract Overview   Thu 22, Apr 2021 - Sun 23, May 2021
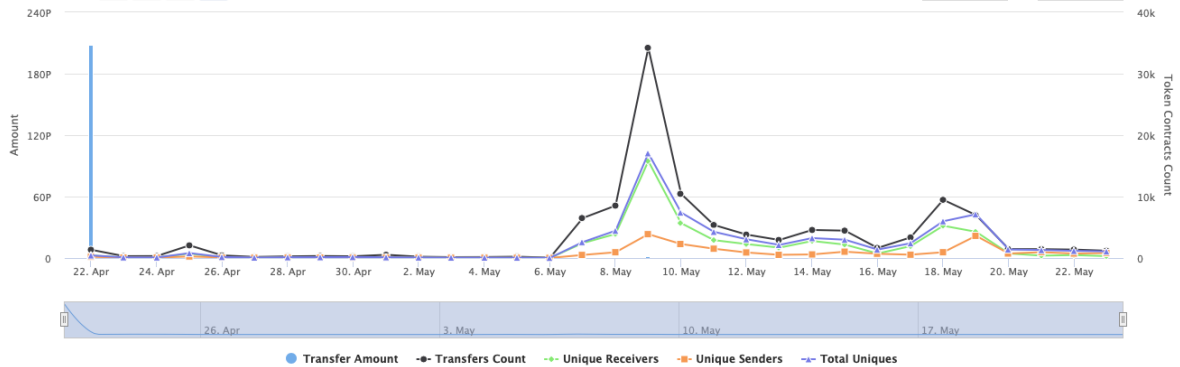
Token Contract 0xd2f9b51c6a1b5598f0ea51ef95d70cb045692d0f (SafeIcarus)
Source: BscScan.com

Zoom 1m 6m 1y All   From Apr 21, 2021 To May 23, 2021

Transfer Amount   Transfers Count   Unique Receivers   Unique Senders   Total Uniques

SafeIcarus top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x0000000000000000000000000000000000000001	98,239,128,916,192,100.620240781	98.2391%
2	📄 PancakeSwap: SAFEICARUS 3	414,275,882,138,294.281893616	0.4143%
3	0x9a4dc72b7f5649943b06820f85a1150056ab9ca0	122,850,887,426,862.847916413	0.1229%
4	0xb2762f44032e3b322d7c5b811dadc61227d46c4e	112,687,744,980,813.360485764	0.1127%
5	0xc7029e939075f48fa2d5953381660c7d01570171	64,375,559,527,865.126863544	0.0644%
6	0xfb4d05a353fbe8a05997a8aa19527116f68f99b3	31,262,570,544,776.232883054	0.0313%
7	0xb738b561402395a47a8e95b4d8933dbfe2e25c12	30,685,684,843,498.25113442	0.0307%
8	0x3134b6a7fd40c2a5b9ce6e0b1f9836f52cf40c24	30,526,359,886,746.358508842	0.0305%
9	0x0b7cd52d3614d0128d7a361853fb0680d1114761	26,127,108,128,520.448676438	0.0261%
10	0x66c45f787fa88964f0980e82872c560fc07eea17	23,461,457,179,741.794501709	0.0235%

SafeIcarus LP token holders

Rank	Address	Quantity	Percentage
1	📄 0x0000000000000000000000000000000000000000	23,875.852133983883413775	92.8308%
2	0x0000000000000000000000000000000000000001	1,732.050807568877292527	6.7343%
3	0x9e2c4933d6228a69149e3011cb1302f3e46a4263	111.859986585776950217	0.4349%

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)
- [Int] <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 **#**

+ **SafeIcarus** (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
- **[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]** 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.

```solidity
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.

```solidity
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 and liquidity fee.

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


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 = now + 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(now > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

Recommendation

❏ Use updated IUniswapV2Router02 instead of older one

Conclusion

Smart contracts contain low severity issues. LP pair contract is not checked.

Liquidity locking details provided by the team by this link -
https://bscscan.com/tx/0x6a893b80c41c0984fecfb6032a0cd270d61d5ba76136eafccf9884cbedbaeb3f

Ownership renounced details provided by the team -
https://bscscan.com/tx/0x15966a424e4939727c3d2479181213665b052379fa767ee9ca10714440303bd0

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.