



TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

June, 2021

Audit Details



Audited project

Black Kishu Inu



Deployer address

0x247C54cE51f8Adf2e5fc1e5025014F23184B8E80



Client contacts:

Black Kishu Inu team



Blockchain

Ethereum



Project website:

<https://bishu.finance>

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 Black Kishu Inu to perform an audit of smart contracts:

<https://etherscan.io/address/0x99043bb680ab9262c7b2ac524e00b215efb7db9b#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 05.06.2021

Contract name	Black Kishu Inu
Contract address	0x99043bB680Ab9262c7b2aC524E00b215Efb7db9b
Total supply	1,000,000,000,000
Token ticker	BISHU
Decimals	9
Token holders	6,110
Transactions count	20,931
Top 100 holders dominance	76.24%
Total fees	231070231603734705324
Uniswap V2 pair	0xfc6f3e19d82868a9386acd23c7118552d04d41e8
Contract deployer address	0x247C54cE51f8Adf2e5fc1e5025014F23184B8E80
Contract's current owner address	0x247c54ce51f8adf2e5fc1e5025014f23184b8e80

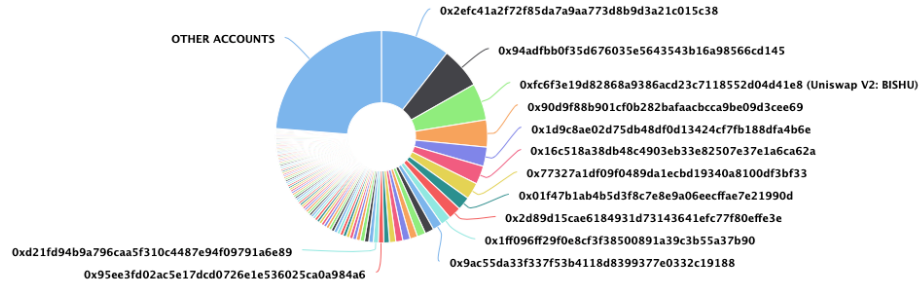
Black Kishu Inu Token Distribution

The top 100 holders collectively own 76.24% (762,429,249,339.14 Tokens) of Black Kishu Inu

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 6,110

Black Kishu Inu Top 100 Token Holders

Source: Etherscan.io



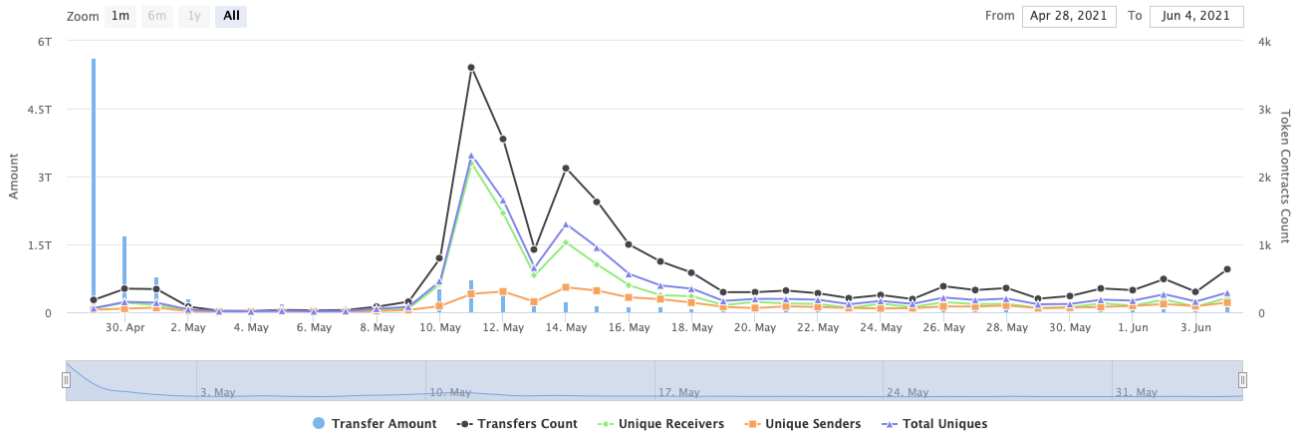
(A total of 762,429,249,339.14 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

Black Kishu Inu Contract Interaction Details


Time Series: Token Contract Overview

Thu 29, Apr 2021 - Fri 4, Jun 2021

Token Contract 0x99043bb680ab9262c7b2ac524e00b215efb7db9b (Black Kishu Inu)
Source: Etherscan.io

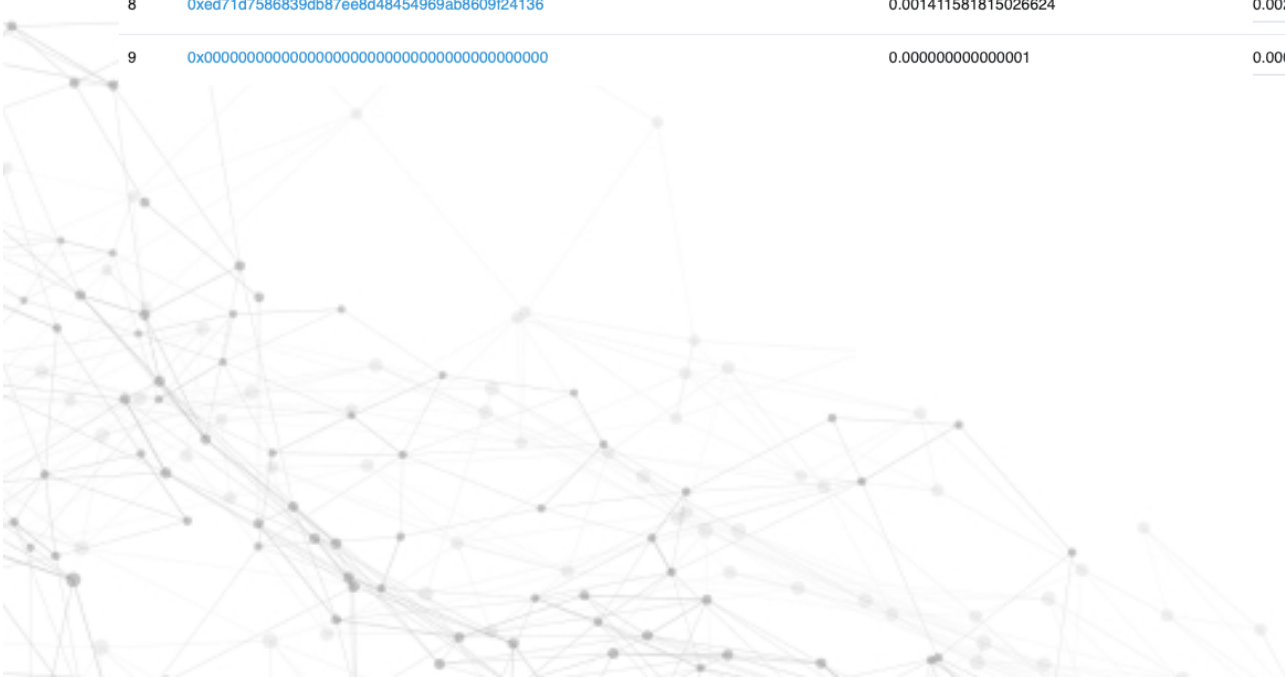


Black Kishu Inu Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x2efc41a2f72f85da7a9aa773d8b9d3a21c015c38	105,567,772,113.130016902	10.5568%
2	0x94adfb0f35d676035e5643543b16a98566cd145	62,579,743,505.823318659	6.2580%
3	 Uniswap V2: BISHU	56,236,509,719.142343022	5.6237%
4	0x90d9f88b901cf0b282bafaacbcc9be09d3cee69	41,377,832,741.512816254	4.1378%
5	0x1d9c8ae02d75db48df0d13424cf7b188dfa4b6e	29,274,602,348.441823943	2.9275%
6	0x16c518a38db48c4903eb33e82507e37e1a6ca62a	28,041,093,323.115587553	2.8041%
7	0x77327a1df09f0489da1ecbd19340a8100df3bf33	25,321,235,620.199299547	2.5321%
8	0x01f47b1ab4b5d3f8c7e8e9a06eecffae7e21990d	20,204,231,332.214285329	2.0204%
9	0x2d89d15cae6184931d73143641efc77f80effe3e	19,546,494,074.97946504	1.9546%
10	0x1ff096ff29f0e8cf3f38500891a39c3b55a37b90	15,731,390,553.303273909	1.5731%

Black Kishu Inu LP Token Holders

Rank	Address	Quantity	Percentage
1	0x6f8fffd641a9228e1b69c35f6e29a60aae2d3f1	34.272880953862509579	49.9771%
2	Vb	15.9999999999999992319	23.3314%
3	Burn Address	15.6227766016838	22.7813%
4	0x503279fd5ee47cd78e65b6773af21bc7722aa68d	2.009560747709655715	2.9304%
5	0x95ee3fd02ac5e17dcd0726e1e536025ca0a984a6	0.51442919545399107	0.7501%
6	0xbcb92712d4b4d99b7cdeed533141fd323820375	0.138277089075408908	0.2016%
7	0xc198054cd7db2edd16bbfe3b1bd03a4bb02c73ec	0.017848140091511576	0.0260%
8	0xed71d7586839db87ee8d48454969ab8609f24136	0.001411581815026624	0.0021%
9	0x00	0.0000000000000001	0.0000%



Contract functions details

+ Context

- [Int] _msgSender
- [Int] _msgData

+ [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

+ [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 #
- + BlackKishu (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] isExcluded
 - [Ext] setExcludeFromFee #
 - modifiers: onlyOwner
 - [Pub] totalFees
 - [Pub] deliver #
 - [Pub] reflectionFromToken
 - [Pub] tokenFromReflection
 - [Ext] excludeAccount #
 - modifiers: onlyOwner
 - [Ext] includeAccount #
 - modifiers: onlyOwner
 - [Prv] removeAllFee #
 - [Prv] restoreAllFee #
 - [Pub] isExcludedFromFee
 - [Prv] _approve #
 - [Prv] _transfer #
 - [Prv] swapTokensForEth #
 - modifiers: lockTheSwap
 - [Prv] sendETHToCharity #
 - [Ext] manualSwap #
 - modifiers: onlyOwner
 - [Ext] manualSend #
 - modifiers: onlyOwner
 - [Ext] setSwapEnabled #
 - modifiers: onlyOwner
 - [Prv] _tokenTransfer #
 - [Prv] _transferStandard #
 - [Prv] _transferToExcluded #
 - [Prv] _transferFromExcluded #
 - [Prv] _transferBothExcluded #
 - [Prv] _takeCharity #
 - [Prv] _reflectFee #
 - [Ext] <Fallback> (\$)
 - [Prv] _getValues
 - [Prv] _getTValues
 - [Prv] _getRValues
 - [Prv] _getRate
 - [Prv] _getCurrentSupply

- [Prv] _getTaxFee
- [Prv] _getMaxTxAmount
- [Pub] _getETHBalance
- [Ext] _setTaxFee #
 - modifiers: onlyOwner
- [Ext] _setCharityFee #
 - modifiers: onlyOwner
- [Ext] _setCharityWallet #
 - modifiers: onlyOwner
- [Ext] _setMaxTxAmount #
 - modifiers: onlyOwner

(\$) = 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 `includeAccount()` 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.

```
fttrace | funcSig
function includeAccount(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.

```
fttrace | funcSig
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:

Check that the excluded array length is not too big.

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

- Owner can change tax fee.

```
ftrace | funcSig
function _setTaxFee(uint256 taxFee↑) external onlyOwner() {
    require(taxFee↑ >= 1 && taxFee↑ <= 10, 'taxFee should be in 1 - 10');
    _taxFee = taxFee↑;
}
```

- Owner can change the maximum transaction amount.

```
ftrace | funcSig
function _setMaxTxAmount(uint256 maxTxAmount↑) external onlyOwner() {
    require(maxTxAmount↑ >= 1000000000000000e9, 'maxTxAmount should be greater than 1000000000000000e9');
    _maxTxAmount = maxTxAmount↑;
}
```

- Owner can exclude from the fee.

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

- Owner can change charity address, charity fee and manually swap and withdraw all ETH balance to charity, moreover half of this amount goes to the marketing.

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

- Owner can lock and unlock. By the way, using these functions the owner could leave as owner 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;
}
```


Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope. Tax and charity fee is private values with no getters, so nobody could their values.

Liquidity locking details 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.



[Techrate1](#)



[Techrate](#)



[Techrate audits](#)