FACTLAB
AUDIT COMPANY

X

July 2021

Security Assessment

ASSESSMENT DETAIL

Audited Project

Karma Inu (KARMA)

Applicant Address

0xef717878841bC89ba1675E88408A7Bd5d6bB232b

Applicant Contact

https://t.me/KarmaInuBSC

Network

Binance Smart Chain (BSC)

Applicant Website

https:// www.karmainubsc.com/

Factlab audit reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. **Factlab** audit do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap, etc'…)

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 **Factlab** and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives)
Factlab audits do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analysed, nor do they provide any indication of the technology proprietors.
Factlab audits should not be used in any way to make decisions around investment or involvement with any particular project.
These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.
Factlab reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk.
Factlab hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against **Factlab**, 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)
Factlab position is that each company and individual are responsible for their own due diligence and continuous security.
Factlab in no way claims any guarantee of security or functionality of the technology we agree to analyze.
The analysis of the security is purely based on the smart contracts alone.

TABLE OF CONTENT

BACKGROUND

Factlab was commissioned by **Karma Inu (KARMA)** to perform an Audit of Smart Contracts:

https://bscscan.com/address/0xef717878841bC89ba1675E88408A7Bd5d6bB232b#code

The purpose of the audit was to achieve the following:

● Ensure that the Smart Contract functions as intended.

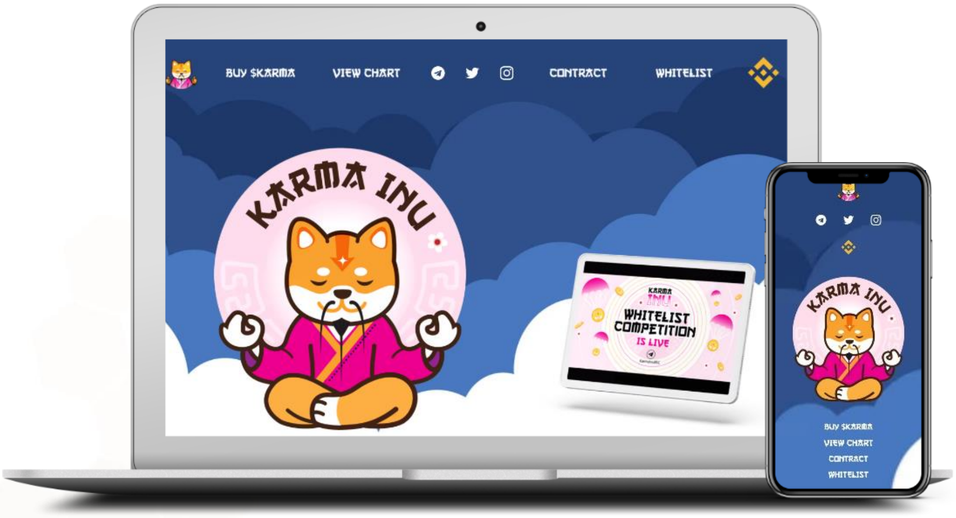
● Identify potential Security issues with the Smart Contract.

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.

WEBSITE OVERVIEW

https://www.karmainubsc.com/

Website Preview & Mobile Test



Above images are actual snapshots of the current live website of the project.

- ✓ Domain was registered on : 10/07/2021, expires on 10/07/2022
- ✓ Responsive to HTML5 & CSS3
- ✓ No severe JavaScript Errors
- ✓ No issues elements loading
- ✓ No Spam / Malware / Popups
- ✓ Spell Check
- ✓ SSL Certificate by R3 | Let’s Encrypt: valid until 9th OCT 2021

Contact Email

marketing@karmainubsc.com

SOCIAL MEDIA OVERVIEW

Official Telegram
https://t.me/KarmaInuBSC

Official Twitter
https://twitter.com/Karma_Inu

Official Instagram
https://www.instagram.com/karmainu/

CONTRACT AUDIT

CONTRACT DETAIL

contract detail as 19.07.2021

Contract Name	Karma Inu
Compiler Version	v0.8.4+commit.c7e474f2
Contract Address	0xef717878841bC89ba1675E88408A7Bd5d6bB232b
Total Supply	2.000.000
Token Symbol	KARMA
Decimals	18
Token Holders	1
Transaction Count	1
Top 100 Holders	100.00%
Liquidity Fee	400
Lottery Fee	50
Total Fees	0
Contract Deployer Address	0x9c6e9d202458775a3640da19436a92f1ed5221c3
Current Owner Address	0x9c6e9d202458775a3640da19436a92f1ed5221c3
uniswapV2Pair	-
uniswapV2Router	-

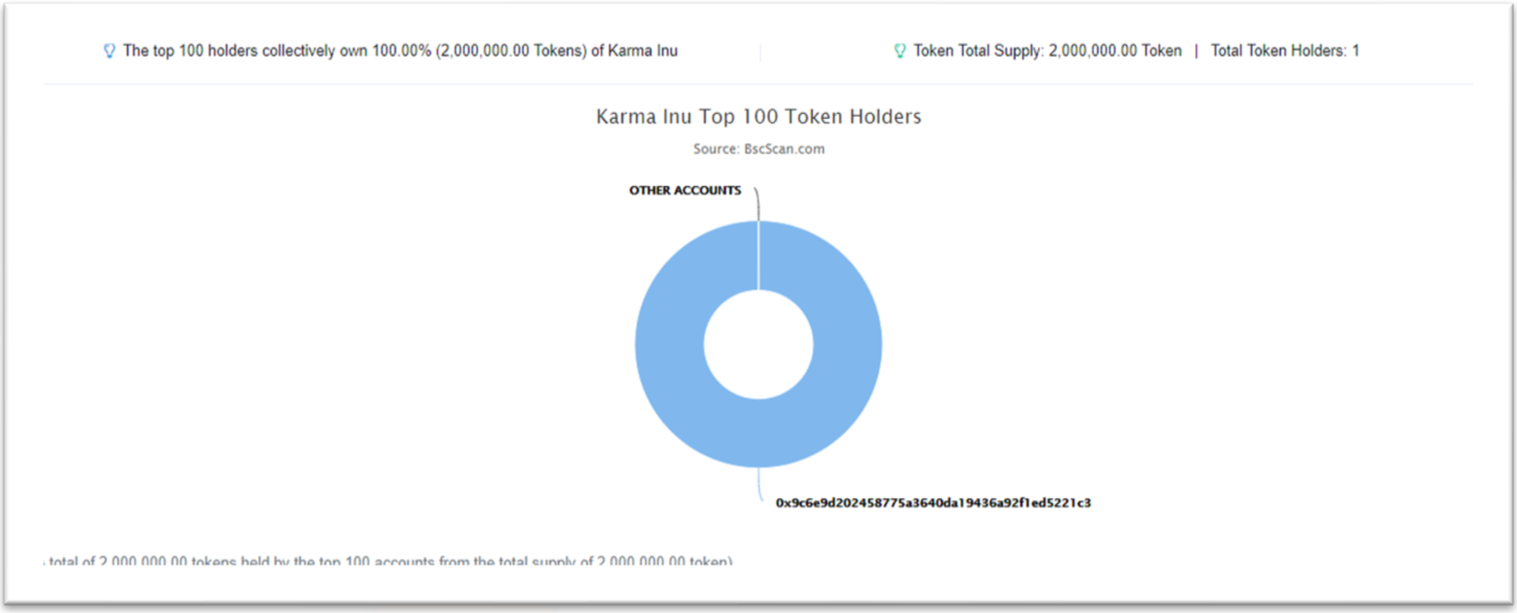
CONTRACT AUDIT

TOKENOMIC DETAILS

Total Supply	2.000.000
Circulating Supply	100.00%
Burned Supply	-
Pre-Sale	-
Marketing/Giveaway wallet	-
Liquidity Locked	Data Not Provided by the Team

CONTRACT AUDIT

TOKEN DISTRIBUTION



The top 100 holders collectively own 100.00% (2,000,000.00 Tokens) of Karma Inu

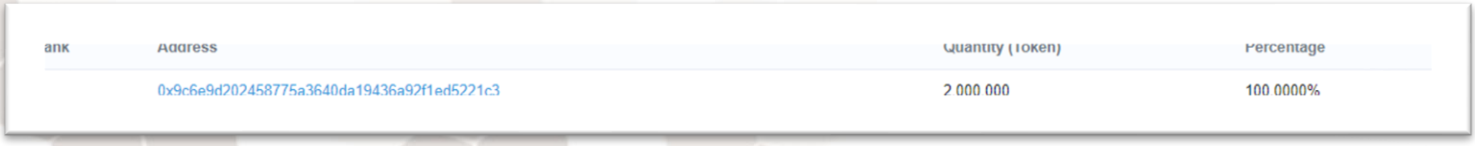
Token Total Supply: 2,000,000.00 Token | Total Token Holders: 1

Karma Inu Top 100 Token Holders

Source: BscScan.com

total of 2,000,000.00 tokens held by the top 100 accounts from the total supply of 2,000,000.00 token)

Karma Inu Top 10 Token Holders

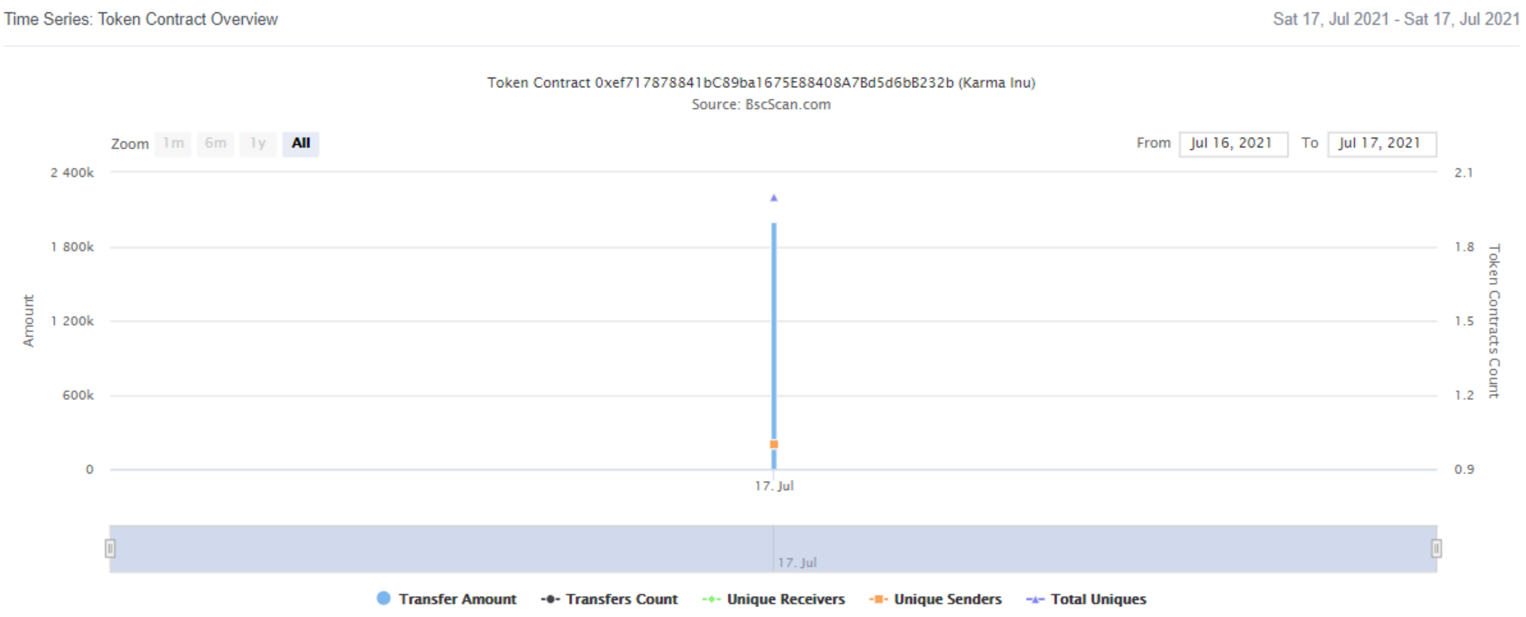


ank	Address	Quantity (Token)	Percentage
	0x9c6e9d202458775a3640da19436a92f1ed5221c3	2,000,000	100.0000%

CONTRACT AUDIT

TOKEN ANALYTICS

Karma Inu Token Contract Interaction Details



Time Series: Token Contract Overview

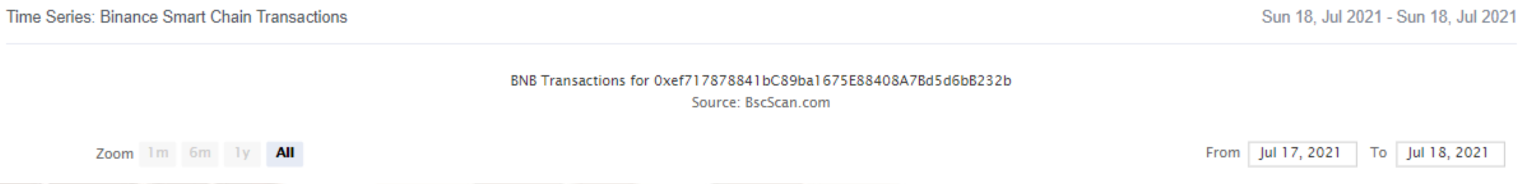
Sat 17, Jul 2021 - Sat 17, Jul 2021

Token Contract 0xef717878841bC89ba1675E88408A7Bd5d6bB232b (Karma Inu)
Source: BscScan.com

Zoom 1m 6m 1y All

From Jul 16, 2021 To Jul 17, 2021

Karma Inu Token Contract Transaction Chart



Time Series: Binance Smart Chain Transactions

Sun 18, Jul 2021 - Sun 18, Jul 2021

BNB Transactions for 0xef717878841bC89ba1675E88408A7Bd5d6bB232b
Source: BscScan.com

Zoom 1m 6m 1y All

From Jul 17, 2021 To Jul 18, 2021

CONTRACT AUDIT

CONTRACT FUNCTIONS DETAIL

```
function _checkLiquidityAdd(address from, address to) private {
    require(!_hasLiqBeenAdded, "Liquidity already added and marked.");
    if (!_hasLimits(from, to) && to == lpPair) {
      _hasLiqBeenAdded = true;
      _liqAddBlock = block.number;
      _liqAddStamp = block.timestamp;

      swapAndLiquifyEnabled = true;
      emit SwapAndLiquifyEnabledUpdated(true);
    }

function setTaxes(uint256 reflectFee,
                uint256 liquidityFee,
                uint256 marketingFee,
                uint256 karmaFee,
                uint256 lotteryFee)
  external onlyOwner() {
    // Fees are set to a master divisor of 10,000. This means 400 fee is 4%, and 40 is 0.4%, etc.
    require(reflectFee <= maxReflectFee
        && liquidityFee <= maxLiquidityFee
        && marketingFee <= maxMarketingFee
        && karmaFee <= maxKarmaFee
        && lotteryFee <= maxLotteryFee
        ); // Prevents owner from abusing fees.
    if (_reflectFee != reflectFee)
      _reflectFee = reflectFee;
    if (_liquidityFee != liquidityFee)
      _liquidityFee = liquidityFee;
```

```
        if (_marketingFee != marketingFee)
            _marketingFee = marketingFee;
        if (_karmaFee != karmaFee)
            _karmaFee = karmaFee;
        if (_lotteryFee != lotteryFee)
            _lotteryFee = lotteryFee;
    }

function allowance(address _owner, address spender) external view returns (uint256);

  /**
   * @dev Sets `amount` as the allowance of `spender` over the caller's tokens.
   *
   * Returns a boolean value indicating whether the operation succeeded.
   *
   * IMPORTANT: Beware that changing an allowance with this method brings the risk
   * that someone may use both the old and the new allowance by unfortunate
   * transaction ordering. One possible solution to mitigate this race
   * condition is to first reduce the spender's allowance to 0 and set the
   * desired value afterwards:
   * https://github.com/ethereum/EIPs/issues/20#issuecomment-263524729
   *
   * Emits an {Approval} event.
   */

function balanceOf(address account) public view override returns (uint256) {
        if (_isExcluded[account]) return _tOwned[account];
        return tokenFromReflection(_rOwned[account]);
    }
```

```
function setBuyBackEnabled(bool _enabled) public onlyOwner {
    buyBackEnabled = _enabled;
    emit BuyBackEnabledUpdated(_enabled);
  }

function setNewRouter(address newRouter) public onlyOwner() {
    IUniswapV2Router02 _newRouter = IUniswapV2Router02(newRouter);
    lpPair = IUniswapV2Factory(_newRouter.factory())
    .createPair(address(this), _newRouter.WETH());
    dexRouter = _newRouter;
  }

function isExcludedFromFee(address account) public view returns(bool) {
    return _isExcludedFromFee[account];
  }

function isExcludedFromReward(address account) public view returns (bool) {
    return _isExcluded[account];
  }

function isSniper(address account) public view returns (bool) {
    return _isSniper[account];
  }
```

```
function reflectionFromToken(uint256 tAmount, bool deductTransferFee) public view
returns(uint256) {
    require(tAmount <= _tTotal, "Amount must be less than supply");
    if (!deductTransferFee) {
      (uint256 rAmount,,,,,) = _getValues(tAmount);
      return rAmount;
    } else {
      (,uint256 rTransferAmount,,,,) = _getValues(tAmount);
      return rTransferAmount;
    }
  }

function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
  }


function tokenFromReflection(uint256 rAmount) public view returns(uint256) {
    require(rAmount <= _rTotal, "Amount must be less than total reflections");
    uint256 currentRate =  _getRate();
    return rAmount.div(currentRate);
  }
```

ISSUES CHECKING STATUS

ISSUE DESCRIPTION	CHECKING STATUS
Compiler Errors	PASSED
Race conditions & Reentrancy - Cross Function Race Conditions	PASSED
Possible Delays in Data Delivery	PASSED
Oracle Calls	PASSED
Timestamp Dependence	PASSED
Integer Overflow and Underflow	PASSED
DoS with Revert	PASSED
DoS with Block Gas Limit	PASSED
Methods Execution Permissions	PASSED
Economy Model of the Contract	PASSED
The Impact of the Exchange Rate on the Logic	PASSED
Private User Data Leaks	PASSED
Malicious Event Log	PASSED
Scooping and Declarations	PASSED
Uninitialized Storage Pointers	PASSED
Arithmetic Accuracy	PASSED
Design Logic	PASSED
Cross Function Race Conditions	PASSED
Safe Open Zeppelin Contracts Implementation & Usage	PASSED
Fallback Function Security	PASSED

ISSUES CHECKING STATUS

SECURITY ISSUES

✓ **High Severity Issues**

No high severity issues found

✓ **Medium Severity Issues**

No medium severity issues found

✓ **Low Severity Issues**

No low severity issues found

Owner privileges

-

Conclusion

Smart contracts contain no severity issues.

Ownership Renounced

-

Liquidity locking details provided by the team:

Data Not Provided by the Team

LP Tokens locked: -

Factlab note:

Please do check Disclaimer Section above and note, **Factlab**'s Audit makes no statements or warranties on any Business Model, Investment Attractiveness or Code Sustainability. The Report is provided for the only as we mentioned in the report and does not include any other potential contracts deployed by Owner.