



TechRate
AUDIT COMPANY

Smart Contract Security Audit

Audit Details



Audited project

GoldenKittyCake



Deployer address

0x6e27844d4d3a040b9c03ee2e9ec7a736dc19853d



Client contacts:

GoldenKittyCake team



Blockchain

Binance Smart Chain



Project website:

goldenkittycake.com

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 GoldenKittyCake to perform an audit of smart contracts:

<https://bscscan.com/address/0x2163913bf7094EC9683401225e7947B698A741ff#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 05.09.2021

| | |
|----------------------------------|--|
| Contract name | GoldenKittyCake |
| Contract address | 0x2163913bf7094EC9683401225e7947B698A741ff |
| Total supply | 1,000,000,000,000 |
| Token ticker | GKCAKE |
| Decimals | 9 |
| Token holders | 1,246 |
| Transactions count | 19,663 |
| Top 100 holders dominance | 82.78% |
| Insurance wallet | 0xc499a2e63d38de517789037e25d69e4fbbc55ef3 |
| Marketing wallet | 0x4d50da60fe164904074a78c82f6024548342b1dc |
| Token pool | 0xb531f3ec0e38e11c09558a0b5c7a52e5c8e2800d |
| Uniswap V2 pair | 0x7f2b0a2fcf37d152ac34ccf03c106952cf26b37d |
| Contract deployer address | 0x6e27844d4d3a040b9c03ee2e9ec7a736dc19853d |
| Contract's current owner address | 0xd40cb4b51830250bff24dd4403e14f11600c197a |

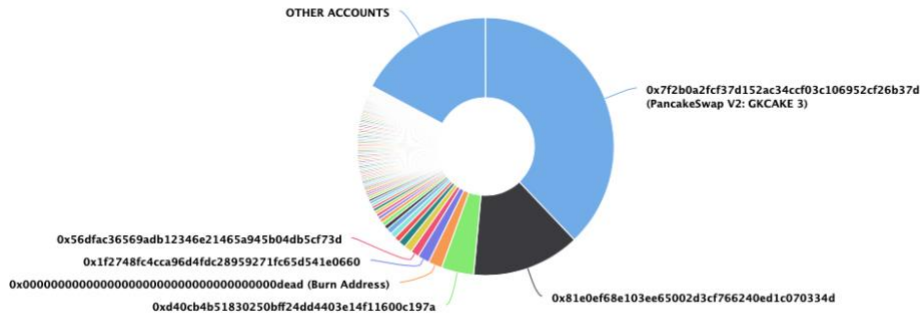
GoldenKittyCake Token Distribution

The top 100 holders collectively own 82.78% (827,796,590,098.99 Tokens) of GoldenKittyCake

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 1,246

GoldenKittyCake Top 100 Token Holders

Source: BscScan.com



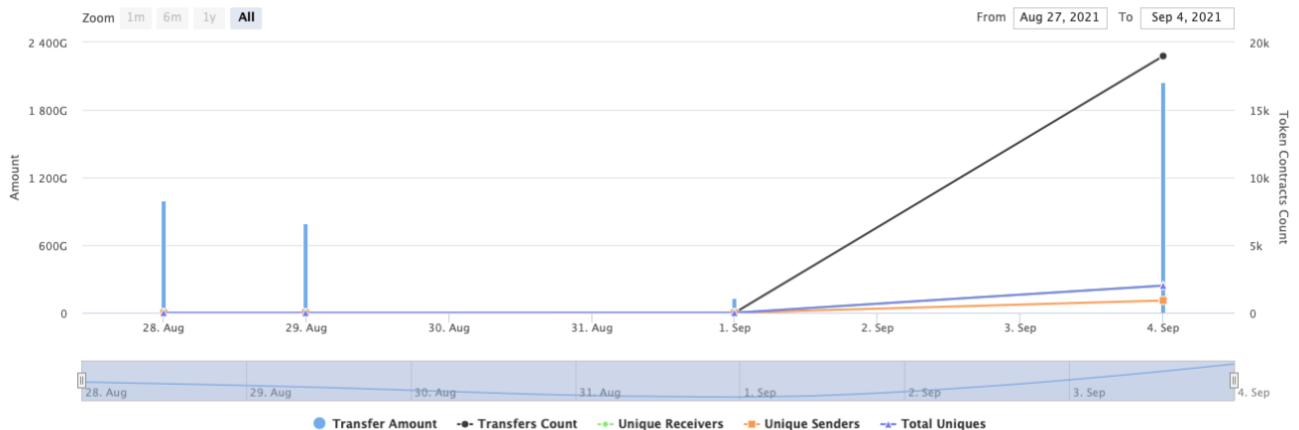
(A total of 827,796,590,098.99 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

GoldenKittyCake Contract Interaction Details





Time Series: Token Contract Overview

Sat 28, Aug 2021 - Sat 4, Sept 2021

Token Contract 0x2163913bf7094EC9683401225e7947B698A741ff (GoldenKittyCake)
Source: BscScan.com



GoldenKittyCake Top 10 Token Holders

| Rank | Address | Quantity (Token) | Percentage |
|------|--|---------------------------|------------|
| 1 |  PancakeSwap V2: GKCAKE 3 | 378,785,983,313.029519684 | 37.8786% |
| 2 |  0x81e0ef68e103ee65002d3cf766240ed1c070334d | 136,065,207,068.992 | 13.6065% |
| 3 | 0xd40cb4b51830250bff24dd4403e14f11600c197a | 40,682,792,931.008 | 4.0683% |
| 4 | Burn Address | 17,545,151,190.433444327 | 1.7545% |
| 5 | 0x1f2748fc4cca96d4fdc28959271fc65d541e0660 | 14,838,294,267.592221713 | 1.4838% |
| 6 |  0x56dfac36569adb12346e21465a945b04db5cf73d | 10,450,668,880.600000059 | 1.0451% |
| 7 | 0x235ea6e206e79b30a630e050e2e47018f999ffa0 | 10,134,924,386.016331361 | 1.0135% |
| 8 |  0xf93c8838c2a6f35cc713eac9740a0fc24f108adf | 9,099,940,381.999999999 | 0.9100% |
| 9 | 0x39d5dcaf772bcb6a05cb3b77659708ee56f33faa | 7,332,000,000 | 0.7332% |
| 10 | 0xe6fa1451903e584dacbf3f0bc928ad0cfce7003d | 7,269,379,714.052579035 | 0.7269% |

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

+ [Int] IUniswapV2Factory

- [Ext] createPair #

+ [Int] IUniswapV2Pair

- [Ext] sync #

+ [Int] IUniswapV2Router01

- [Ext] factory
- [Ext] WETH
- [Ext] addLiquidity #
- [Ext] addLiquidityETH (\$)

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)

- [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
 - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
 - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
 - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens (\$)
- + ReentrancyGuard
- [Pub] <Constructor> #
- + [Lib] TransferHelper
- [Int] safeApprove #
 - [Int] safeTransfer #
 - [Int] safeTransferFrom #
 - [Int] safeTransferETH #
- + Wallet
- [Ext] <Fallback> (\$)
- + GoldenKittyCake (Context, IERC20, Ownable, ReentrancyGuard)
- [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
 - [Pub] reflectionFromToken
 - [Pub] tokenFromReflection
 - [Ext] excludeAccount #
 - modifiers: onlyOwner
 - [Ext] includeAccount #
 - modifiers: onlyOwner
 - [Prv] _approve #
 - [Prv] _transfer #
 - [Prv] collectFee #
 - [Pub] calculateReward
 - [Pub] claimReward #
 - modifiers: isHuman,nonReentrant,lockTheSwap
 - [Prv] topUpClaimCycleAfterTransfer #
 - [Pub] calculateTopUpClaim
 - [Prv] _getReflectionRate
 - [Prv] swap #
 - modifiers: lockTheSwap
 - [Prv] swapAndLiquify #
 - [Prv] swapTokensForCake #
 - [Prv] swapTokensForEth #
 - [Prv] swapEthForTokens #
 - [Prv] addLiquidity #
 - [Ext] setPairRouterCake #
 - modifiers: onlyOwner

- [Ext] setTaxless #
 - modifiers: onlyOwner
- [Ext] setSwapEnabled #
 - modifiers: onlyOwner
- [Ext] setFeeActive #
 - modifiers: onlyOwner
- [Ext] setTaxFee #
 - modifiers: onlyOwner
- [Ext] setInsuranceFee #
 - modifiers: onlyOwner
- [Ext] setMarketingFee #
 - modifiers: onlyOwner
- [Ext] setLiquidityFee #
 - modifiers: onlyOwner
- [Ext] setMarketingWallet #
 - modifiers: onlyOwner
- [Ext] setInsuranceWallet #
 - modifiers: onlyOwner
- [Ext] setMaxTxAmount #
 - modifiers: onlyOwner
- [Ext] setMinTokensBeforeSwap #
 - modifiers: onlyOwner
- [Ext] setRewardCycleInterval #
 - modifiers: onlyOwner
- [Ext] <Fallback> (\$)

(\$)= 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(!_excluded[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:

Check that the excluded array length is not too big.

Notes:

- swapEthForTokens is private and unused.

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

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

```
ftrace | funcSig
function setTaxFee(uint256 buy↑, uint256 sell↑, uint256 p2p↑) external onlyOwner {
    _taxFee[0] = buy↑;
    _taxFee[1] = sell↑;
    _taxFee[2] = p2p↑;
}

ftrace | funcSig
function setInsuranceFee(uint256 buy↑, uint256 sell↑, uint256 p2p↑) external onlyOwner {
    _insuranceFee[0] = buy↑;
    _insuranceFee[1] = sell↑;
    _insuranceFee[2] = p2p↑;
}

ftrace | funcSig
function setMarketingFee(uint256 buy↑, uint256 sell↑, uint256 p2p↑) external onlyOwner {
    _marketingFee[0] = buy↑;
    _marketingFee[1] = sell↑;
    _marketingFee[2] = p2p↑;
}

ftrace | funcSig
function setLiquidityFee(uint256 buy↑, uint256 sell↑, uint256 p2p↑) external onlyOwner {
    _liqFee[0] = buy↑;
    _liqFee[1] = sell↑;
    _liqFee[2] = p2p↑;
}
```

- Owner can enable/disable swap.

```
function setSwapEnabled(bool enabled↑) external onlyOwner {
    swapEnabled = enabled↑;
    SwapUpdated(enabled↑);
}
```

- Owner can change the maximum transaction amount.

```
ftrace | funcSig
function setMaxTxAmount(uint256 amount↑) external onlyOwner {
    maxTxAmount = amount↑;
}
```

- Owner can change reward cycle interval.

```
function setRewardCycleInterval(uint256 interval↑) external onlyOwner {
    rewardCycleInterval = interval↑;
}
```

- Owner can change pair, router and cake addresses.

```
function setPairRouterCake(  
    address _pair↑,  
    IUniswapV2Router02 _router↑,  
    address _cake↑  
) external onlyOwner {  
    pair = _pair↑;  
    router = _router↑;  
    cake = _cake↑;  
}
```

- Owner can exclude from the taxes.

```
ftrace | funcSig  
function setTaxless(address account↑, bool value↑) external onlyOwner {  
    isTaxless[account↑] = value↑;  
}
```

- Owner can disable and enable fees.

```
ftrace | funcSig  
function setFeeActive(bool value↑) external onlyOwner {  
    isFeeActive = value↑;  
}
```

- Owner can change marketing and insurance wallets.

```
ftrace | funcSig  
function setMarketingWallet(address wallet↑) external onlyOwner {  
    marketingWallet = wallet↑;  
}  
  
ftrace | funcSig  
function setInsuranceWallet(address bank↑) external onlyOwner {  
    insuranceWallet = bank↑;  
}
```

- Owner can change minimum amount of tokens needed to swap.

```
ftrace | funcSig  
function setMinTokensBeforeSwap(uint256 amount↑) external onlyOwner {  
    minTokensBeforeSwap = amount↑;  
}
```

Conclusion

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

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](#)