Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

October, 2021

Audit Details

Audited project

Gatsby Inu

Deployer address

0x4f3a8b848d186ab81ec0656959c3ff771c381d0b

Client contacts:

Gatsby Inu team

Blockchain

Ethereum

Project website:

Not provided

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 Gatsby Inu to perform an audit of smart contracts:

https://etherscan.io/address/0xd60d8e670438615721c8f50db31839f98a124ff7#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

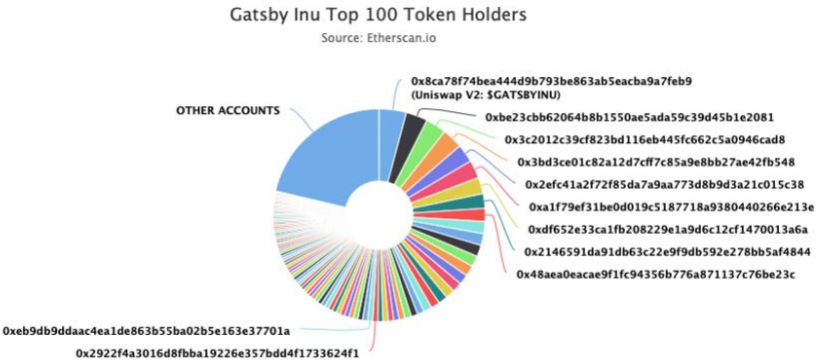
Token contract details for 29.10.2021

Contract name	Gatsby Inu
Contract address	0xD60d8E670438615721c8F50db31839f98A124Ff7
Total supply	1,000,000,000,000
Token ticker	$GATSBYINU
Decimals	9
Token holders	555
Transactions count	2,371
Top 100 holders dominance	78.69%
Marketing fee	9
Tax fee	0
Total fees	0
Uniswap V2 pair	0x8ca78f74bea444d9b793be863ab5eacba9a7feb9
Contract deployer address	0x4f3a8b848d186ab81ec0656959c3ff771c381d0b
Contract's current owner address	0x0000000000000000000000000000000000000000

Gatsby Inu Token Distribution

The top 100 holders collectively own 78.69% (786,883,452,898.53 Tokens) of Gatsby Inu

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 555

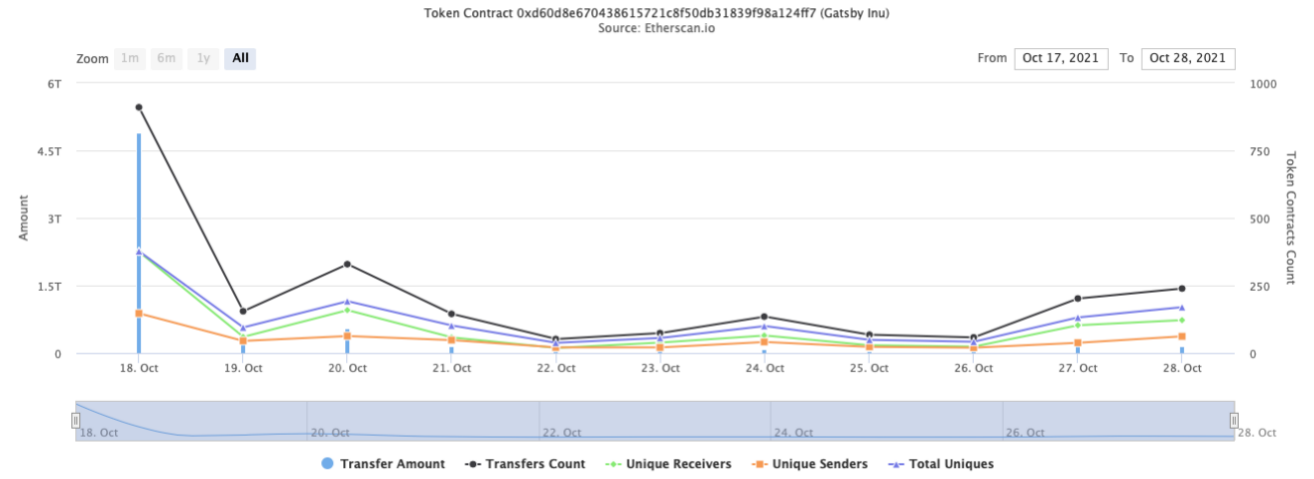


(A total of 786,883,452,898.53 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

Gatsby Inu Contract Interaction Details

Time Series: Token Contract Overview

Mon 18, Oct 2021 - Thu 28, Oct 2021



Gatsby Inu Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	Uniswap V2: $GATSBYINU	41,676,736,403.813566824	4.1677%
2	0xbe23cbb62064b8b1550ae5ada59c39d45b1e2081	32,802,039,806.604235533	3.2802%
3	0x3c2012c39cf823bd116eb445fc662c5a0946cad8	31,419,678,432.917965885	3.1420%
4	0x3bd3ce01c82a12d7cff7c85a9e8bb27ae42fb548	31,416,904,470.73980491	3.1417%
5	0x2efc41a2f72f85da7a9aa773d8b9d3a21c015c38	27,743,408,369.169463501	2.7743%
6	0xa1f79ef31be0d019c5187718a9380440266e213e	27,553,304,873.223275888	2.7553%
7	0xdf652e33ca1fb208229e1a9d6c12cf1470013a6a	24,826,000,000	2.4826%
8	0x2146591da91db63c22e9f9db592e278bb5af4844	22,283,989,008.19	2.2284%
9	0x48aea0eacae9f1fc94356b776a871137c76be23c	19,883,669,586.600781	1.9884%
10	0xeb467f831233c47b25877eaf895773c6031d7e71	18,200,000,000	1.8200%

Contract functions details

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Lib] SafeMath
  - [Int] tryAdd
  - [Int] trySub
  - [Int] tryMul
  - [Int] tryDiv
  - [Int] tryMod
  - [Int] add
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] mod
  - [Int] sub
  - [Int] div
  - [Int] mod

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

+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Int] functionStaticCall
  - [Int] functionStaticCall
  - [Int] functionDelegateCall #
  - [Int] functionDelegateCall #
  - [Prv] _verifyCallResult

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner

+ [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 #

+ GatsbyInu (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] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward #
  - modifiers: onlyOwner
- [Ext] includeInReward #
  - modifiers: onlyOwner
- [Pub] excludeFromFee #
  - modifiers: onlyOwner
- [Pub] includeInFee #
  - modifiers: onlyOwner
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Ext] <Fallback> ($)
- [Prv] _reflectFee #
- [Ext] addToBlackList #
  - modifiers: onlyOwner
- [Ext] removeFromBlackList #
  - modifiers: onlyOwner
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeMarketing #
- [Prv] calculateTaxFee
- [Prv] calculateMarketingFee
- [Pub] isExcludedFromFee
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] setFees #
- [Prv] SwapAndSend #

    - modifiers: lockTheSwap
  - [Prv] _tokenTransfer #
  - [Prv] _transferStandard #
  - [Prv] _transferToExcluded #
  - [Prv] _transferFromExcluded #
  - [Prv] _transferBothExcluded #
  - [Ext] setDefaultMarketingFee #
    - modifiers: onlyOwner
  - [Ext] setMarketingFee4Sellers #
    - modifiers: onlyOwner
  - [Pub] setFeesOnSellersAndBuyers #
    - modifiers: onlyOwner
  - [Pub] setSwapAndSendEnabled #
    - modifiers: onlyOwner
  - [Pub] setnumTokensToExchangeForMarketing #
    - modifiers: onlyOwner
  - [Ext] _setMarketingWallet #
    - 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 `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:

Check that the excluded array length is not too big.

- The function `addToBlackList()` uses the loop to add addresses from function argument to blacklist. It also could be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

```
function addToBlackList(address[] calldata addresses⬆) external onlyOwner {
 for (uint256 i; i < addresses⬆.length; ++i) {
   _isBlacklisted[addresses⬆[i]] = true;
 }
}
```

Recommendation:

Check that the array length is not too big.

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

- Owner can change fees.

```
function setDefaultMarketingFee(uint256 marketingFee⬆) external onlyOwner() {
    defaultMarketingFee = marketingFee⬆;
}

ftrace | funcSig
function setMarketingFee4Sellers(uint256 marketingFee4Sellers⬆) external onlyOwner() {
    _marketingFee4Sellers = marketingFee4Sellers⬆;
}

ftrace | funcSig
function setFeesOnSellersAndBuyers(bool _enabled⬆) public onlyOwner() {
    feesOnSellersAndBuyers = _enabled⬆;
}
```

- Owner can change the maximum transaction amount.

```
function _setMaxTxAmount(uint256 maxTxAmount⬆) external onlyOwner() {
    _maxTxAmount = maxTxAmount⬆;
}
```

- Owner can remove addresses from blacklist.

```
function removeFromBlackList(address account⬆) external onlyOwner {
    _isBlacklisted[account⬆] = false;
}
```

- Owner can exclude from the fee.

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

- Owner can change number of tokens to exchange for marketing.

```
function setnumTokensToExchangeForMarketing(uint256 _numTokensToExchangeForMarketing⬆) public onlyOwner() {
    numTokensToExchangeForMarketing = _numTokensToExchangeForMarketing⬆;
}
```

- Owner can change marketing wallet.

```
function _setMarketingWallet(address payable wallet⬆) external onlyOwner() {
    marketingWallet = wallet⬆;
}
```

Conclusion

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

Liquidity locking details are provided by the team:
https://dxsale.app/app/v3/dxlockview?id=0&add=0x4F3a8b848d186ab81Ec0656959C3ff771c381d0B&type=lplock&chain=ETH

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.