



Smart Contract Security Audit

Audit details:

Audited project:	GorillaDiamond
Deployer address:	0x287fe17da81d9881703f75ec3ab21e2e2adf7afd
Client contacts:	GorillaDiamond team
Blockchain:	Binance Smart Chain
Project website:	https://gorilladiamond.com

April, 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 GorillaDiamond to perform an audit of smart contracts:

- <https://bscscan.com/address/0xb7f2bca9b034f8cc143339dd12bb31d3d50cf27a#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 18.04.2021.

Contract name:	GorillaDiamond
Contract address:	0xb7f2bca9b034f8cc143339dd12bb31d3d50cf27a
Total supply:	1_000_000_000_000_000_000_000_000
Token ticker:	GDT
Decimals:	9
Token holders:	2713
Transactions count:	10862
Top 100 holders dominance:	95.76 %
Contract deployer address:	0x287fe17da81d9881703f75ec3ab21e2e2adf7afd
Contract's current owner address:	0x287fe17da81d9881703f75ec3ab21e2e2adf7afd
Current tax fee:	6 percent
Current liquidity fee:	5 percent
Total fees:	28_538_063_430_669_056_513_608
Max tx amount:	1_000_000_000_000_000_000_000_000
Total reflections:	112_534_278_952_407_229_289_637_164_158_802_021_276_750_560_284_201_015_069_691_131_571_036_077_774_186
Uniswap V2 router:	0x05ff2b0db69458a0750badebc4f9e13add608c7f
Uniswap V2 pair:	0x11cef9a89e8eccff1598fa429b095630bea6c5d2

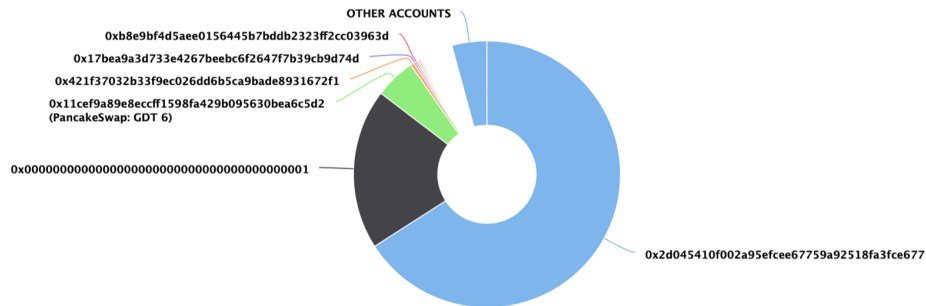
GorillaDiamond token distribution

💡 The top 100 holders collectively own 95.76% (957,565,240,989,424.00 Tokens) of GorillaDiamond

💡 Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 2,713

GorillaDiamond Top 100 Token Holders

Source: BscScan.com



(A total of 957,565,240,989,424.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

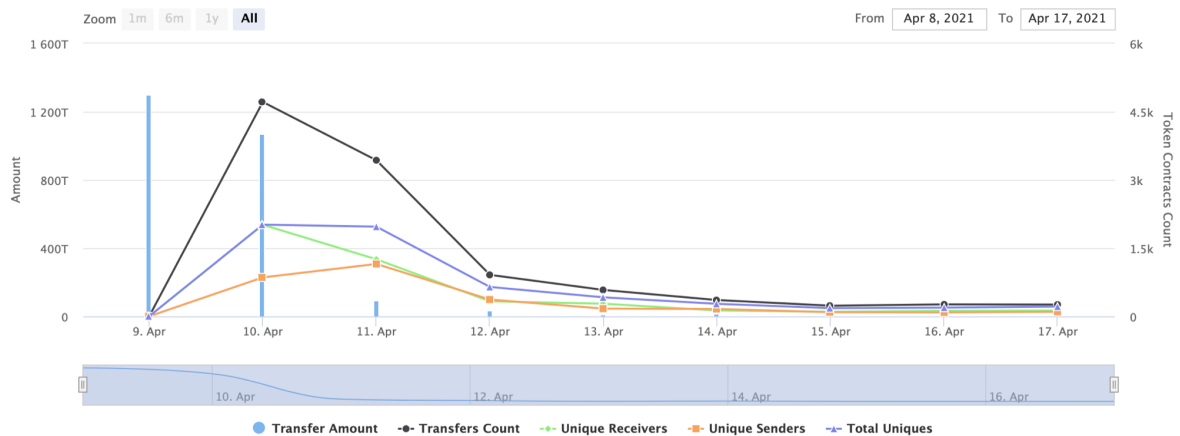
GorillaDiamond contract interaction details

Time Series: Token Contract Overview



Fri 9, Apr 2021 - Sat 17, Apr 2021

Token Contract 0xb7f2bca9b034f8cc143339dd12bb31d3d50cf27a (GorillaDiamond)

Source: BscScan.com



GorillaDiamond top 10 token holders

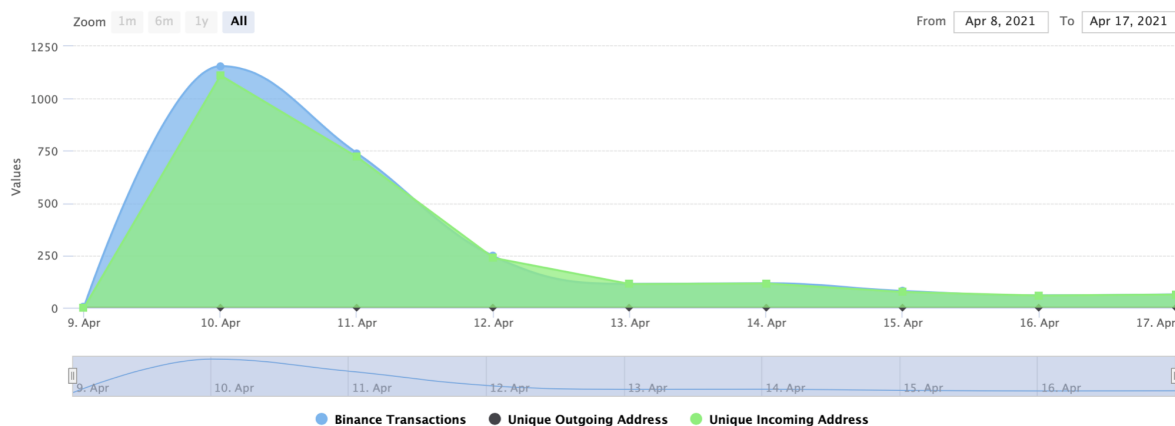
Rank	Address	Quantity (Token)	Percentage
1	 0x2d045410f002a95efcee67759a92518fa3fce677	659,241,008,700,000.027	65.9241%
2	0x0001	194,679,768,420,344.470715684	19.4680%
3	 PancakeSwap: GDT 6	49,208,533,459,313.786926408	4.9209%
4	0x421f37032b33f9ec026dd6b5ca9bade8931672f1	4,416,977,411,843.317982077	0.4417%
5	0x17bea9a3d733e4267beebc6f2647f7b39cb9d74d	2,249,104,272,200.754216551	0.2249%
6	0xb8e9bf4d5ae0156445b7bdbb2323f2cc03963d	2,082,469,222,646.241138769	0.2082%
7	0xfa6f87c348efb94c022bd5279fb7fa631a904101	2,059,685,629,068.053660338	0.2060%
8	0x7758ae53336a8b60db3bc3d2be2d476a8ffff6f9e	1,738,712,700,052.020768281	0.1739%
9	0x0689b8ec16ae09db3b64e46dc9cb21a9bc285a61	1,591,034,641,743.199518194	0.1591%
10	0x061d10dd8dbb38f93f35207ce0c3110c8722a240	1,391,608,839,314.786731816	0.1392%

GorillaDiamond transactions

Time Series: Binance Smart Chain Transactions

Fri 9. Apr 2021 - Sat 17. Apr 2021

BNB Transactions for 0xb7f2bca9b034f8cc143339dd12bb31d3d50cf27a
Source: BscScan.com



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

+ 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] createPair #

+ [Int] IUniswapV2Router

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

+ GorillaDiamond (Context, IERC20, Ownable)

- [Pub] <Constructor> #
- [Pub] increaseAllowance #
- [Pub] decreaseAllowance #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward #
 - modifiers: onlyOwner
- [Ext] includeInReward #
 - modifiers: onlyOwner
- [Pub] balanceOf
- [Pub] excludeFromFee #
 - modifiers: onlyOwner
- [Pub] includeInFee #
 - modifiers: onlyOwner
- [Ext] setTaxFeePercent #
 - modifiers: onlyOwner
- [Ext] setLiquidityFeePercent #
 - modifiers: onlyOwner
- [Ext] setMaxTxPercent #
 - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
 - modifiers: onlyOwner
- [Prv] _getTokenValues
- [Int] _getReflectionValues
- [Int] _getCurrentSupply
- [Int] _getRate
- [Int] _getValues
- [Pub] deliver #
- [Int] _approve #
- [Pub] approve #
- [Int] _swapTokensForEth #
- [Int] _addLiquidity #
- [Int] _swapAndLiquify #
 - modifiers: lockTheSwap
- [Int] _transferStandard #
- [Int] _tokenTransfer #
- [Int] _transfer #
- [Pub] transfer #
- [Pub] transferFrom #
- [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(!isExcludedFromReward(account), "Account is already excluded");

    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            tokensOwned[account] = 0;
            isExcludedFromReward(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() internal view returns (uint256 reflectionSupply, uint256 tokenSupply) {
    reflectionSupply = totalReflections;
    tokenSupply = totalSupply;

    for (uint256 i = 0; i < _excluded.length; i++) {
        if (reflectionsOwned[_excluded[i]] > reflectionSupply || tokensOwned[_excluded[i]] > tokenSupply) return (totalReflections, totalSupply);

        reflectionSupply = reflectionSupply.sub(reflectionsOwned[_excluded[i]]);
        tokenSupply = tokenSupply.sub(tokensOwned[_excluded[i]]);
    }

    return reflectionSupply < totalReflections.div(totalSupply) ? (totalReflections, totalSupply) : (reflectionSupply, tokenSupply);
}
```

Recommendation:

Use `EnumerableSet` instead of array or do not use long arrays.

Owner privileges

- ☐ Owner can change the tax fee.
- ☐ Owner can change the liquidity fee.
- ☐ Owner can change the max tx amount.

Conclusion

Smart contracts do not contain high severity issues.

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.