TechRate
September, 2022

SMART CONTRACTS SECURITY AUDIT REPORT

Techrate_audits Techrate Techrate1

Audit Details

Audited project

BeerusShiba

Deployer address

0xa96458281a5e019a319fdf922bb17dbc105dcd3a

Client contacts:

BeerusShiba team

Blockchain

Binance Smart Chain

Project website:

https://www.beerusshiba.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 BeerusShiba to perform an audit of smart contracts:
https://bscscan.com/address/0xa0408Df6846591c9Fb9980c96aed01FC1817eE97#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 28.09.2022

Contract name	BeerusShiba
Contract address	0xa0408Df6846591c9Fb9980c96aed01FC1817eE97
Total supply	1,000,000,000,000,000
Token ticker	BSHIB
Decimals	9
Token holders	475
Transactions count	2,053
Top 100 holders dominance	78.44%
Percent marketing	50
Buy fee	8
Sell fee	8
Uniswap V2 pair	0x400ffd2f74ff389ad8a22275264ee06707f2c6a2
Contract deployer address	0xa96458281a5e019a319fdf922bb17dbc105dcd3a
Owner address	0x0000000000000000000000000000000000000000

BeerusShiba Token Distribution

The top 100 holders collectively own 78.44% (784,380,440,651,276.00 Tokens) of BeerusShiba

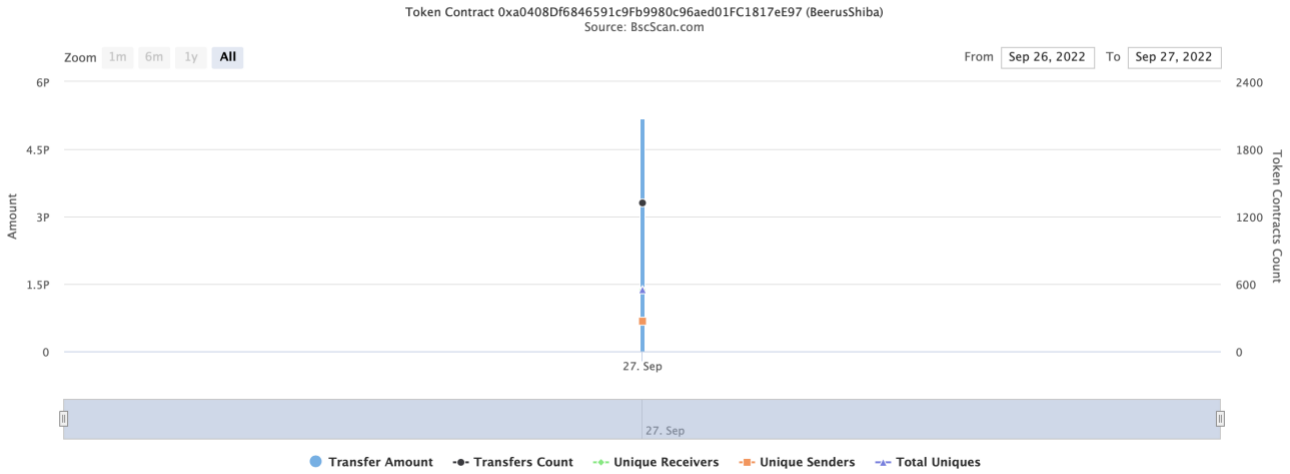
Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 475

(A total of 784,380,440,651,276.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

BeerusShiba Contract Interaction Details

Time Series: Token Contract Overview

Tue 27, Sept 2022 - Tue 27, Sept 2022



BeerusShiba Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: BSHIB 198	119,009,066,745,146.457248023	11.9009%
2	0xdca61c0eca6b214432e247e2b333471ac527fc6e	19,123,234,572,111.20492787	1.9123%
3	0xefb317c867be62fe2b992464c47b30aad7a3da0f	18,415,058,555,870.22415123	1.8415%
4	0x0dde964da34747c11e16656af1c547cd9bdb7d0b	18,307,207,765,999.252460934	1.8307%
5	0xb0252181cd211521b65113ab85b846182ef684a4	16,905,915,400,000.736	1.6906%
6	0xd2b26dcd6e8be6435f98a025266c8ccc1f998aeb	15,759,633,612,814.65546969	1.5760%
7	0xe0d7ac8979b94fd998961e1049f646beb7b5984c	15,577,461,973,733.215541144	1.5577%
8	0x08af2be6e1e92d6f612106dfff474bb1b482e79	15,327,042,723,829.817213682	1.5327%
9	0xc90af8ae3bafab82052726318aa58f53a3272415	15,039,585,294,592.406031512	1.5040%
10	0x28ae01fb49a5af809ce1f1050e4dcfc1f25ff333	14,626,069,708,564.336669054	1.4626%

Contract functions details

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

+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] sub
  - [Int] div

+ 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

+ [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] 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 #

+  BeerusShiba (Context, IERC20)
  - [Pub] owner
  - [Pub] renounceOwnership #
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Ext] <Fallback> ($)
  - [Prv] _getCurrentSupply
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] sendToWallet #
  - [Prv] swapAndLiquify #
    - modifiers: lockTheSwap
  - [Prv] swapTokensForBNB #
  - [Prv] addLiquidity #
  - [Pub] remove_Random_Tokens #
  - [Prv] _tokenTransfer #


($) = 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.**	Passed
10.	**Methods execution permissions.**	Passed
11.	**Economy model of the contract.**	High issues
12.	**The impact of the exchange rate on the logic.**	Passed
13.	**Private user data leaks.**	Passed
14.	**Malicious Event log.**	Low issues
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

1. Burn issue

Issue:

- With each transfer to a burning wallet, its balance increases, but the total supply decreases. After a certain number of such transactions, the sum of the balances will not be equal to the total supply.

Recommendation:
Revise burn logic of the contract and keep only one way of burning – decreasing total supply, or sending tokens to zero address.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

2. Event emitting

Issue:

- With each transfer to a burning wallet, its balance increases, but the total supply decreases. And there is no emitting of decreasing total supply event.

Recommendation:
Revise burn logic of the contract and add event emitting.

Testnet deployment

Contracts Description Table

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
BeerusShiba	Implementation	Context, IERC20		
└	transfer	Public ❗	🛑	NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	remove_Random_Tokens	Public ❗	🛑	NO❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Conclusion

Smart contracts contain high severity issues! Liquidity pair contract's security is not checked due to out of scope. The further transfers and operations with the funds raise are not related to this particular contract.

Liquidity locking details are provided by the team:
https://mudra.website/?certificate=yes&type=0&lp=0x400ffd2f74ff389ad8a22275264ee06707f2c6a2

Ownership renounce details are provided by the team:
https://bscscan.com/tx/0xcd4c7d81832a329a68018a2a79dd7f762801bae0bb7ce6abbc085de84b91e130

Security score: 67.

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.