# RugFreeCoins Audit

# EJECT Token

## Smart Contract Security Audit

June 16, 2021

# Contents

# Audit details

**Audited project**

EJECT Token

**Contract Address**

0x2d43Dfe648Bb8C9c22f9FFcD1937369a9Fdd0eBc

**Client contact**

EJECT Token Team

**Blockchain**

Binance smart chain

**Project website**

https://www.eject.space/

# 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 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 Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins 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 Rugfreecoins 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, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, 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

Rugfreecoins was commissioned by EJECT Token to perform an audit of the smart contract.

**https://bscscan.com/address/0x2d43Dfe648Bb8C9c22f9FFcD1937369a9Fdd0eBc**

The focus of this audit is to verify that the smart contract is secure, resilient and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, long term sustainability and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

# About the project

The EJECT Token was created on the 20th of May 2021, and held a private sale on the 26th, the presale happendon the 28th on the platform DxSale, and launched on PancakeSwap on the 30th of May to be publicly traded. Initially called “EJECTElon” and it has now been rebranded to EJECT.

EJECT aims to become the first crowdfunding platform for upcoming projects on the Binance Smart Chain. The expected date for the launch is in late Q3 2021. the EJECT team is working to develop a marketplace where investors can meet and find legit projects with actual and strong use cases. The EJECT team also aims to support any project developers from the inception of a project to its launch through the platform. EJECT will host the most promising upcoming projects on the BSC and will offer the guarantee that all projects that make it to the platform are safe to invest in.

## Current tokenomics

- **2%** of token trade goes holders pockets.
- **2%** of token trades goes to liquidity pool.

## Future tokenomics (planned)

- **0%** of token trade will go to holders pockets.
- **3%** of token trades will go to liquidity pool.

# Target market and the concept

## Target market

- Project developers from BSC projects with long term sustainable use cases.
- Investors to find legit, long term projects.
- Long term Eject token holders to benefit from the platform growth and traction.
- New crypto entries to be educated about crypto through the platform.
- Traders to balance the market.

## Core concept

- A platform to find legit projects with strong use cases for the investors in a very early stage.
- A platform to list projects and reach potential long-term investors for the developers.

EJECT platform is aiming to create a marketspace for the new crypto projects through a screening mechanism, that helps the projects with strong use cases to get recognized and get the visibility that they deserve regardless of the marketing, and helps the community to select the good legit projects.

## EJECT business model

### I. How will EJECT generate revenue?

Taking a small percentage of commission from the projects that the EJECT team assists and helps to sustain, and through the advertisements.

### II. Token Buyback

25% of the company’s annual profits will serve to buy circulating tokens and burn them. This will decrease the supply every year without impacting the price.

### III. How will holders benefit from EJECT success?

When the platform grows and sustains in the industry, EJECT token value will eventually grow day by day. The share of the investors will also grow higher in the long run.

## How the platform works

1. **Screening project application**

**a)** Project application submission with detailed project action plan, goals, long term plans and the budget strategies through the EJECT platform. The EJECT team will review them and assist project devs in terms of marketing and project launches.

**b)** The EJECT team will classify the project into categories.

- i. Meme/heavily marketed projects with no real use-case.
- ii. Projects with promising ideas and use-cases.
- iii. Hybrids with varying levels of marketing/value added.

**c)** Developing teams will have to formulate a deadline for raising funds for their project.

2. **Assisting with contract creation and submission (optional service)**

The EJECT team will assist the projects to code the contract aligning with the special project use cases. When coding the token, the EJECT team will guarantee that the contract is Rug-Proof and that it includes no “backdoors” or functionalities that might endanger potential holders. This will increase the legitimacy of the projects that are listed in our platform.

Tokens that are coded externally will have to go through a rigorous auditing process prior to being launched, which can be handled by the EJECT team or any third-party service.

3. **The Crowdfunding section: Allowing people to invest in projects**

**a)** Investors will be able to invest in the project of their choice. They will be able to see how much each project is trying to raise and invest their desired amount into that specific project. Investing in projects on the EJECT platform will be possible with **BNB** and the **$EJECT token**. The choice will be at the discretion of developers. Choosing $EJECT will yield significant advantages to the developers, as they will benefit from much lower fees, and exclusive extra assistance in making their project a success.

**b)** These projects will feature in our day's releases and their ratings will be decided based on the below criteria;

- i. Number of contributors.
- ii. Total value backed.
- iii. Number of video plays and the % of video watched.
- iv. Click throughs.

**c)** The step-by-step funding unlock feature.

The EJECT team will set up milestones and KPIs for the projects and unlock the funds upon reaching the milestones step by step. This will happen within the system itself and the development teams will have to submit payment proofs that will be available in public. The step-by-step funding unlock feature guarantees that all of the early investors' money goes in the right places.

**d)** Rewarding investors

Investors will be rewarded with project's tokens that they selected, but through a carefully predetermined mechanism to keep the market healthy.

- Early investors will be awarded proportionally higher since the risk of getting in early is higher.
- A mechanism of locking the token purchases to avoid dumping the whole amount at once, which will damage the market and eventually the project.

**e)** Project launch and the development

i. Make developers' identity known to the EJECT team to ensure the legitimacy of the project.

ii. Set specific milestones of the projects and the progress will be publicly visible in the EJECT platform.

iii. Launching the Token will be done through the EJECT team, via a fully integrated platform within EJECT (similar to DxSale, Unicrypt, etc.). The EJECT team will require ownership of the contract through launch. in order to guarantee that everything is done safely. EJECT will keep ownership of the contracts we code and ask for a transfer of ownership of pre-written contracts. After launch, the EJECT team will transfer the ownership. The EJECT team will guarantee that contracts do not include any feature that can potentially be harmful to investors.

## Additional services

1. For developers, EJECT will initiate and facilitate contact with influencers, and guarantee that the money is spent where it should be. EJECT team help with other marketing expenses and processes that might be unfamiliar to developing teams with lack of experience in that field.
2. For the community to communicate with developers, EJECT will have a TALK WITH THE TEAMS section on the platform, where developers can schedule voice and video chats with investors, or communicate through messages.

3. For investors that do NOT wish to invest money directly into a project, EJECT platform will offer the opportunity to “shill”, or help in developing the project itself. Compensation for these efforts will be distributed in EJECT tokens that have been raised for the project and considered as a Marketing or developing expense. No ‘project token’ will be rewarded for these tasks. Other rewards, including merch, will also be given to top investors.

4. The distribution of tokens after the end of the funding period and prior to launch of the project.

5. The partnership with wholesale companies to create merch and distribute to people who complete tasks for a project.

## Safety features EJECT offers

- Project identity checks and screening process.
- Audit requirements and ownership transfer feature.
- Early investor lock feature.
- Step by step unlock feature.
- A required liquidity lock of 100 years for all projects’ tokens that are launched through EJECT platform.
- Constant monitoring of activities from our EJECT team, and direct communication with developing teams.
- A monitored dev wallet, locked for 1 year, and with the same progressive unlocking feature as large pre-launch investors.
- The breakdown of projects through crowdfunding page into 2 sections: Doxxed and non-Doxxed.

## Potential to grow with score points

| | | |
|---|---|---|
| 1. | Project efficiency | 9/10 |
| 2. | Project uniqueness | 9/10 |
| 3 | Information quality | 8/10 |
| 4 | Service quality | 9/10 |
| 5 | System quality | 8/10 |
| 6 | Impact on the community | 10/10 |
| 7 | Impact on the business | 9/10 |
| 8 | Preparing for the future | 8/10 |
| Total Points | | **8.75/10** |

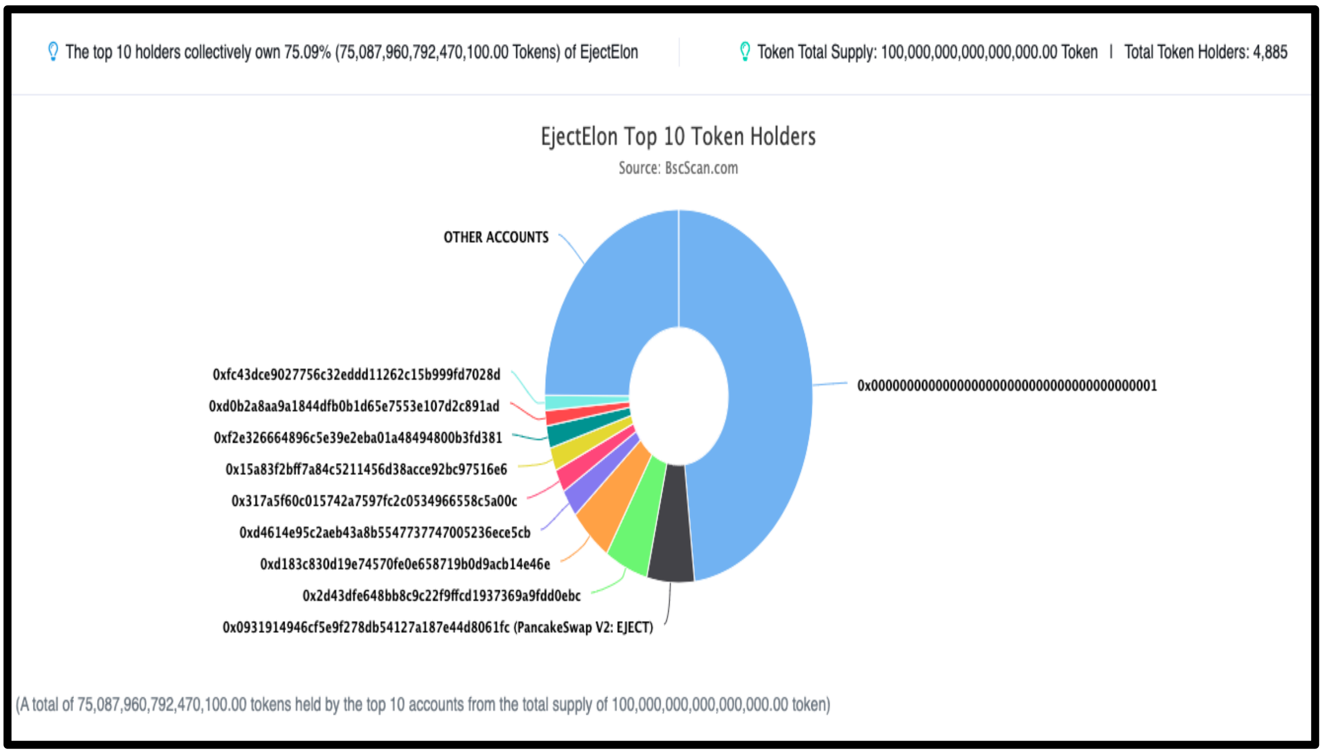
# Contract details

## Token contract details for 16th June 2021

| | |
|---|---|
| **Contract name** | EjectElon |
| **Contract address** | 0x2d43Dfe648Bb8C9c22f9FFcD1937369a9Fdd0eBc |
| **Token supply** | 100,000,000,000,000,000 |
| **Token ticker** | EJECT |
| **Decimals** | 9 |
| **Token holders** | 4,694 |
| **Transaction count** | 23,614 |
| **Top 100 holders dominance** | 91.30% |
| **Contract deployer address** | 0x168026A30ce69C3313A97CB3d519829C03E72438 |
| **Contract’s current owner address** | 0x9f75cd5865736253d9c37f82b3c3e200938d96c4 |

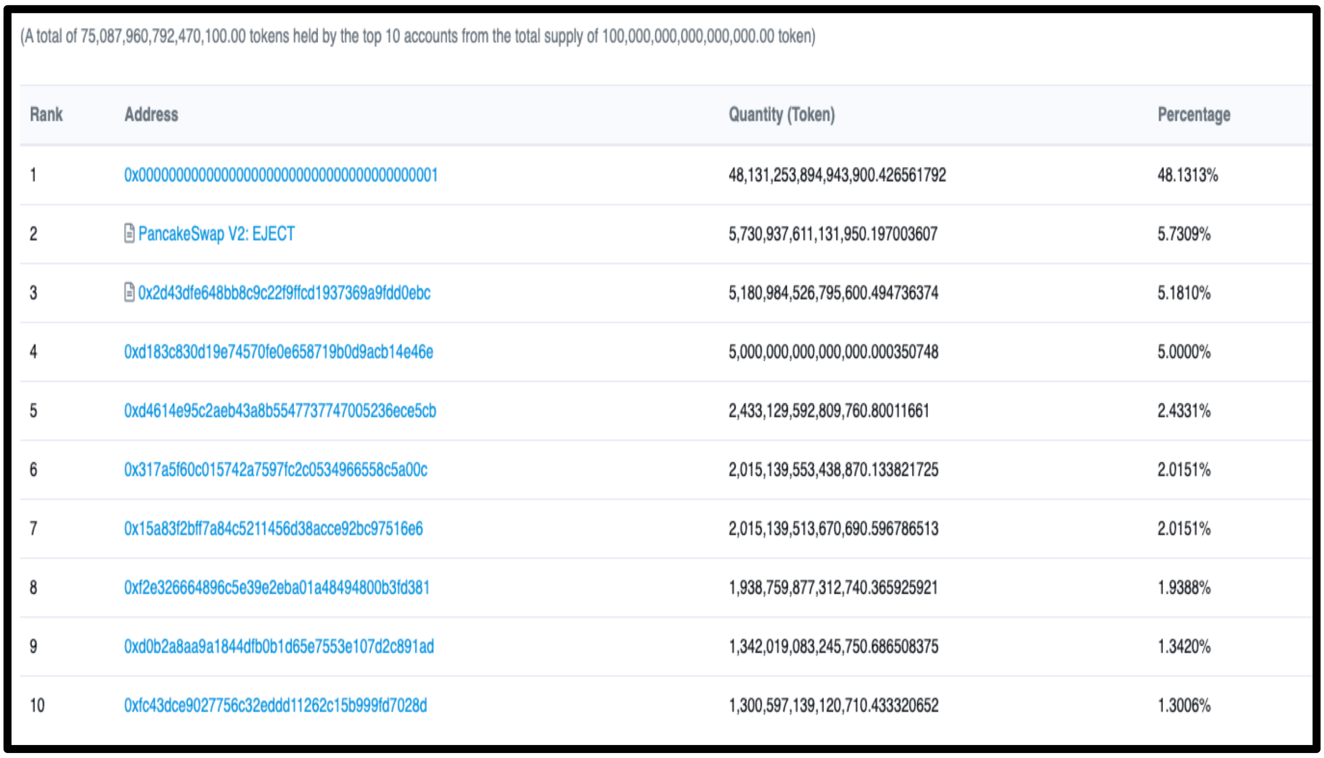
# Top token holders

## Top 10 Token Holders as at 16th June 2021



The top 10 holders collectively own 75.09% (75,087,960,792,470,100.00 Tokens) of EjectElon | Token Total Supply: 100,000,000,000,000,000.00 Token | Total Token Holders: 4,885

(A total of 75,087,960,792,470,100.00 tokens held by the top 10 accounts from the total supply of 100,000,000,000,000,000.00 token)



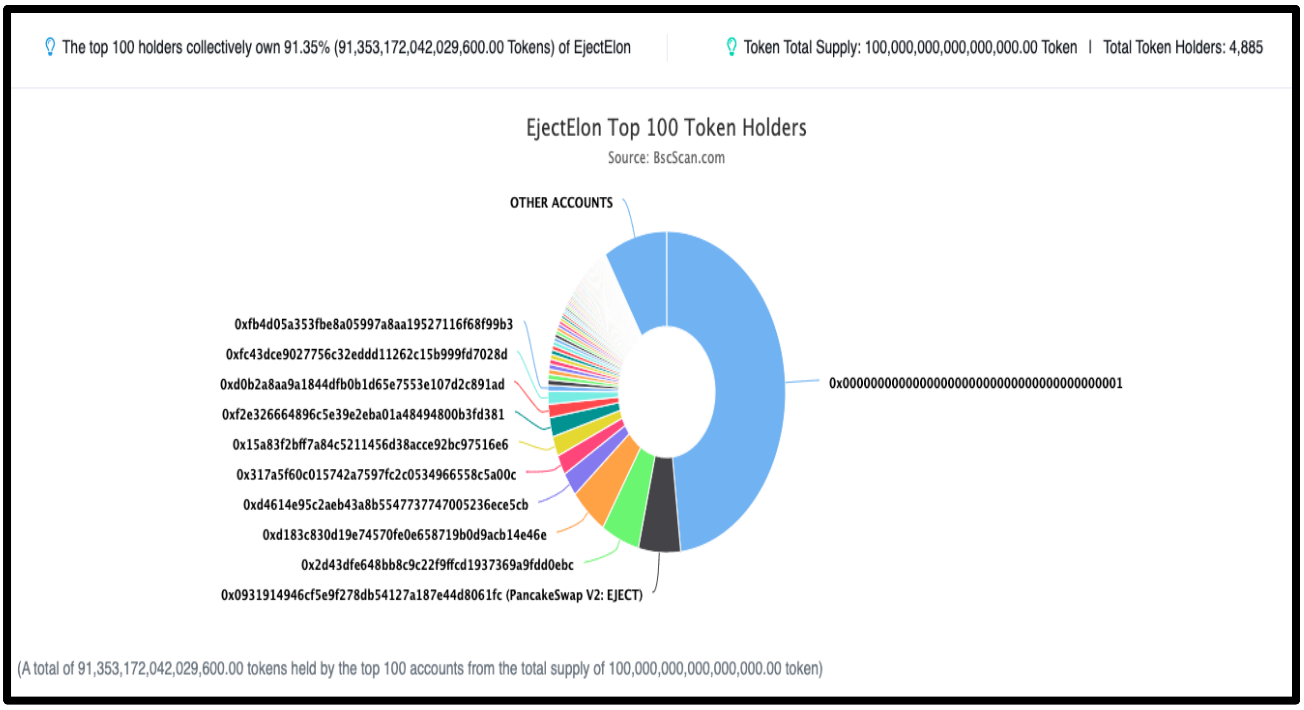
(A total of 75,087,960,792,470,100.00 tokens held by the top 10 accounts from the total supply of 100,000,000,000,000,000.00 token)

| Rank | Address | Quantity (Token) | Percentage |
|---|---|---|---|
| 1 | 0x0000000000000000000000000000000000000001 | 48,131,253,894,943,900.426561792 | 48.1313% |
| 2 | PancakeSwap V2: EJECT | 5,730,937,611,131,950.197003607 | 5.7309% |
| 3 | 0x2d43dfe648bb8c9c22f9ffcd1937369a9fdd0ebc | 5,180,984,526,795,600.494736374 | 5.1810% |
| 4 | 0xd183c830d19e74570fe0e658719b0d9acb14e46e | 5,000,000,000,000,000.000350748 | 5.0000% |
| 5 | 0xd4614e95c2aeb43a8b5547737747005236ece5cb | 2,433,129,592,809,760.80011661 | 2.4331% |
| 6 | 0x317a5f60c015742a7597fc2c0534966558c5a00c | 2,015,139,553,438,870.133821725 | 2.0151% |
| 7 | 0x15a83f2bff7a84c5211456d38acce92bc97516e6 | 2,015,139,513,670,690.596786513 | 2.0151% |
| 8 | 0xf2e326664896c5e39e2eba01a48494800b3fd381 | 1,938,759,877,312,740.365925921 | 1.9388% |
| 9 | 0xd0b2a8aa9a1844dfb0b1d65e7553e107d2c891ad | 1,342,019,083,245,750.686508375 | 1.3420% |
| 10 | 0xfc43dce9027756c32eddd11262c15b999fd7028d | 1,300,597,139,120,710.433320652 | 1.3006% |

# Token distribution

**Token will be distributed as follows:**

## Top 100 Token Holders as at 16th June 2021


The top 100 holders collectively own 91.35% (91,353,172,042,029,600.00 Tokens) of EjectElon

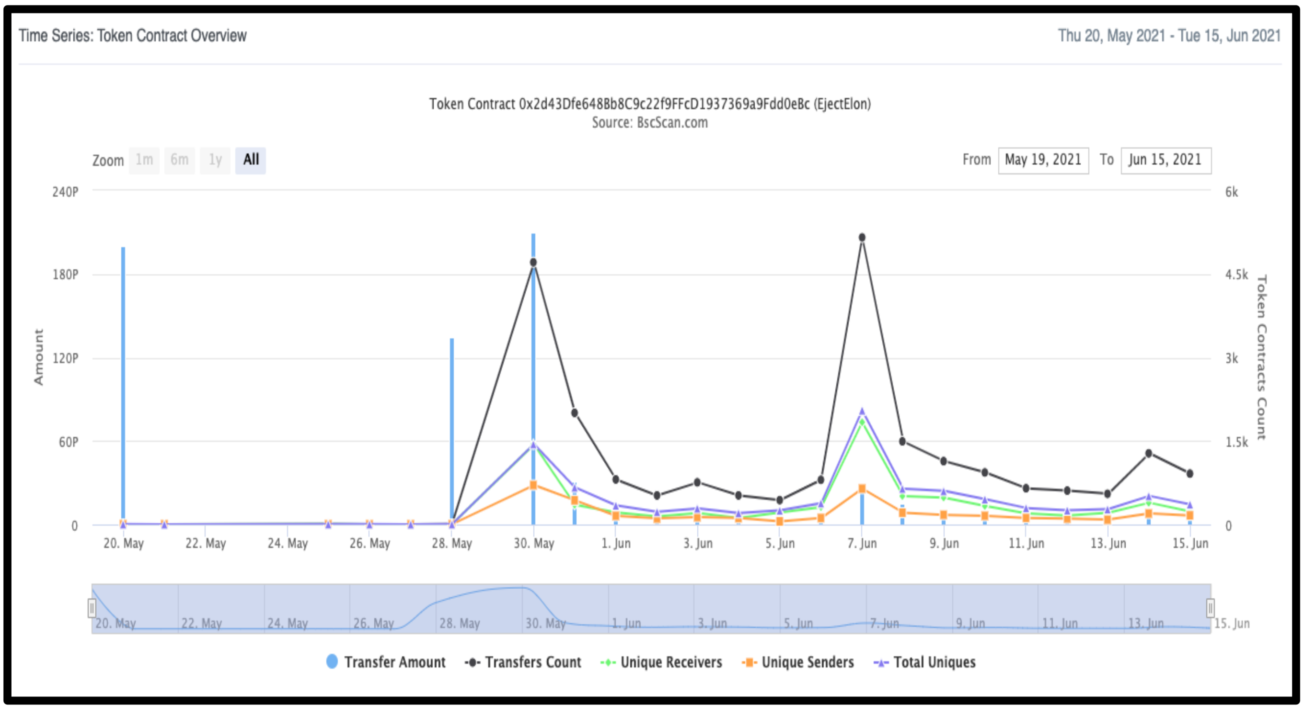
Token Total Supply: 100,000,000,000,000,000.00 Token | Total Token Holders: 4,885

EjectElon Top 100 Token Holders

Source: BscScan.com

(A total of 91,353,172,042,029,600.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

# Contract interaction details


Time Series: Token Contract Overview

Thu 20, May 2021 - Tue 15, Jun 2021

Token Contract 0x2d43Dfe648Bb8C9c22f9FFcD1937369a9Fdd0eBc (EjectElon)

Source: BscScan.com

# Contract code function details

| No | Category | Item | Result |
|---|---|---|---|
| 1 | Coding conventions | BRC20 Token standards | pass |
| | | compile errors | pass |
| | | Compiler version security | pass |
| | | visibility specifiers | pass |
| | | Gas consumption | low issue |
| | | SafeMath features | pass |
| | | Fallback usage | pass |
| | | tx.origin usage | pass |
| | | deprecated items | pass |
| | | Redundant code | pass |
| | | Overriding variables | pass |
| 2 | Function call audit | Authorization of function call | pass |
| | | Low level function (call/delegate call) security | pass |
| | | Returned value security | pass |
| | | Selfdestruct function security | pass |
| 3 | Business security | Access control of owners | pass |
| | | Business logics | pass |
| | | Business implementations | pass |
| 4 | Integer overflow/underflow | | pass |
| 5 | Reentrancy | | pass |
| 6 | Exceptional reachable state | | pass |
| 7 | Transaction ordering dependence | | pass |
| 8 | Block properties dependence | | pass |
| 9 | Pseudo random number generator (PRNG) | | pass |
| 10 | DoS (Denial of Service) | | pass |
| 11 | Token vesting implementation | | pass |
| 12 | Fake deposit | | pass |
| 13 | Event security | | pass |

# Contracts description table

Below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions and implementations with its visibility and mutability.

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| **IERC20** | **Interface** | | | |
| L | totalSupply | External ❗ | | NO❗ |
| L | balanceOf | External ❗ | | NO❗ |
| L | transfer | External ❗ | 🛑 | NO❗ |
| L | allowance | External ❗ | | NO❗ |
| L | approve | External ❗ | 🛑 | NO❗ |
| L | transferFrom | External ❗ | 🛑 | NO❗ |
| | | | | |
| **SafeMath** | **Library** | | | |
| L | add | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | mul | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | mod | Internal 🔒 | | |
| L | mod | Internal 🔒 | | |
| | | | | |

| **Context** | **Implementation** | | | |
|---|---|---|---|---|
| L | _msgSender | Internal 🔒 | | |
| L | _msgData | Internal 🔒 | | |
| | | | | |
| **Address** | **Library** | | | |
| L | isContract | Internal 🔒 | | |
| L | sendValue | Internal 🔒 | 🛑 | |
| L | functionCall | Internal 🔒 | 🛑 | |
| L | functionCall | Internal 🔒 | 🛑 | |
| L | functionCallWith Value | Internal 🔒 | 🛑 | |
| L | functionCallWith Value | Internal 🔒 | 🛑 | |
| L | _functionCallWit hValue | Private 🔐 | 🛑 | |
| | | | | |
| **Ownable** | **Implementation** | **Context** | | |
| L | | Internal 🔒 | 🛑 | |
| L | owner | Public ❗ | | NO❗ |
| L | renounceOwner ship | Public ❗ | 🛑 | onlyOwner |
| L | transferOwnersh ip | Public ❗ | 🛑 | onlyOwner |
| L | geUnlockTime | Public ❗ | | NO❗ |
| L | lock | Public ❗ | 🛑 | onlyOwner |
| L | unlock | Public ❗ | 🛑 | NO❗ |
| | | | | |
| **IUniswapV2Factory** | **Interface** | | | |
| L | feeTo | External ❗ | | NO❗ |

| | | | | |
|---|---|---|---|---|
| └ | feeToSetter | External ❗ | | NO❗ |
| └ | getPair | External ❗ | | NO❗ |
| └ | allPairs | External ❗ | | NO❗ |
| └ | allPairsLength | External ❗ | | NO❗ |
| └ | createPair | External ❗ | 🛑 | NO❗ |
| └ | setFeeTo | External ❗ | 🛑 | NO❗ |
| └ | setFeeToSetter | External ❗ | 🛑 | NO❗ |
| | | | | |
| **IUniswapV2Pair** | **Interface** | | | |
| └ | name | External ❗ | | NO❗ |
| └ | symbol | External ❗ | | NO❗ |
| └ | decimals | External ❗ | | NO❗ |
| └ | totalSupply | External ❗ | | NO❗ |
| └ | balanceOf | External ❗ | | NO❗ |
| └ | allowance | External ❗ | | NO❗ |
| └ | approve | External ❗ | 🛑 | NO❗ |
| └ | transfer | External ❗ | 🛑 | NO❗ |
| └ | transferFrom | External ❗ | 🛑 | NO❗ |
| └ | DOMAIN_SEPARATOR | External ❗ | | NO❗ |
| └ | PERMIT_TYPEHASH | External ❗ | | NO❗ |
| └ | nonces | External ❗ | | NO❗ |
| └ | permit | External ❗ | 🛑 | NO❗ |
| └ | MINIMUM_LIQUIDITY | External ❗ | | NO❗ |
| └ | factory | External ❗ | | NO❗ |
| └ | token0 | External ❗ | | NO❗ |

| | | | | |
|---|---|---|---|---|
| └ | token1 | External ❗ | | NO❗ |
| └ | getReserves | External ❗ | | NO❗ |
| └ | price0CumulativeLast | External ❗ | | NO❗ |
| └ | price1CumulativeLast | External ❗ | | NO❗ |
| └ | kLast | External ❗ | | NO❗ |
| └ | mint | External ❗ | 🛑 | NO❗ |
| └ | burn | External ❗ | 🛑 | NO❗ |
| └ | swap | External ❗ | 🛑 | NO❗ |
| └ | skim | External ❗ | 🛑 | NO❗ |
| └ | sync | External ❗ | 🛑 | NO❗ |
| └ | initialize | External ❗ | 🛑 | NO❗ |
| | | | | |
| **IUniswapV2Router01** | **Interface** | | | |
| └ | factory | External ❗ | | NO❗ |
| └ | WETH | External ❗ | | NO❗ |
| └ | addLiquidity | External ❗ | 🛑 | NO❗ |
| └ | addLiquidityETH | External ❗ | 💵 | NO❗ |
| └ | removeLiquidity | External ❗ | 🛑 | NO❗ |
| └ | removeLiquidityETH | External ❗ | 🛑 | NO❗ |
| └ | removeLiquidityWithPermit | External ❗ | 🛑 | NO❗ |
| └ | removeLiquidityETHWithPermit | External ❗ | 🛑 | NO❗ |
| └ | swapExactTokensForTokens | External ❗ | 🛑 | NO❗ |
| └ | swapTokensForExactTokens | External ❗ | 🛑 | NO❗ |

| | | | | |
|---|---|---|---|---|
| L | swapExactETHForTokens | External ❗ | 💵 | NO❗ |
| L | swapTokensForExactETH | External ❗ | 🛑 | NO❗ |
| L | swapExactTokensForETH | External ❗ | 🛑 | NO❗ |
| L | swapETHForExactTokens | External ❗ | 💵 | NO❗ |
| L | quote | External ❗ | | NO❗ |
| L | getAmountOut | External ❗ | | NO❗ |
| L | getAmountIn | External ❗ | | NO❗ |
| L | getAmountsOut | External ❗ | | NO❗ |
| L | getAmountsIn | External ❗ | | NO❗ |
| | | | | |
| **IUniswapV2Router02** | **Interface** | **IUniswapV2Router01** | | |
| L | removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 | NO❗ |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| | | | | |

| EjectElon | Implementation | Context, IERC20, Ownable | | |
|---|---|---|---|---|
| L | | Public ❗ | 🛑 | NO❗ |
| L | name | Public ❗ | | NO❗ |
| L | symbol | Public ❗ | | NO❗ |
| L | decimals | Public ❗ | | NO❗ |
| L | totalSupply | Public ❗ | | NO❗ |
| L | balanceOf | Public ❗ | | NO❗ |
| L | transfer | Public ❗ | 🛑 | NO❗ |
| L | allowance | Public ❗ | | NO❗ |
| L | approve | Public ❗ | 🛑 | NO❗ |
| L | transferFrom | Public ❗ | 🛑 | NO❗ |
| L | increaseAllowance | Public ❗ | 🛑 | NO❗ |
| L | decreaseAllowance | Public ❗ | 🛑 | NO❗ |
| L | isExcludedFromReward | Public ❗ | | NO❗ |
| L | totalFees | Public ❗ | | NO❗ |
| L | deliver | Public ❗ | 🛑 | NO❗ |
| L | reflectionFromToken | Public ❗ | | NO❗ |
| L | tokenFromReflection | Public ❗ | | NO❗ |
| L | excludeFromReward | Public ❗ | 🛑 | onlyOwner |
| L | includeInReward | External ❗ | 🛑 | onlyOwner |
| L | _transferBothExcluded | Private 🔐 | 🛑 | |
| L | excludeFromFee | Public ❗ | 🛑 | onlyOwner |
| L | includeInFee | Public ❗ | 🛑 | onlyOwner |

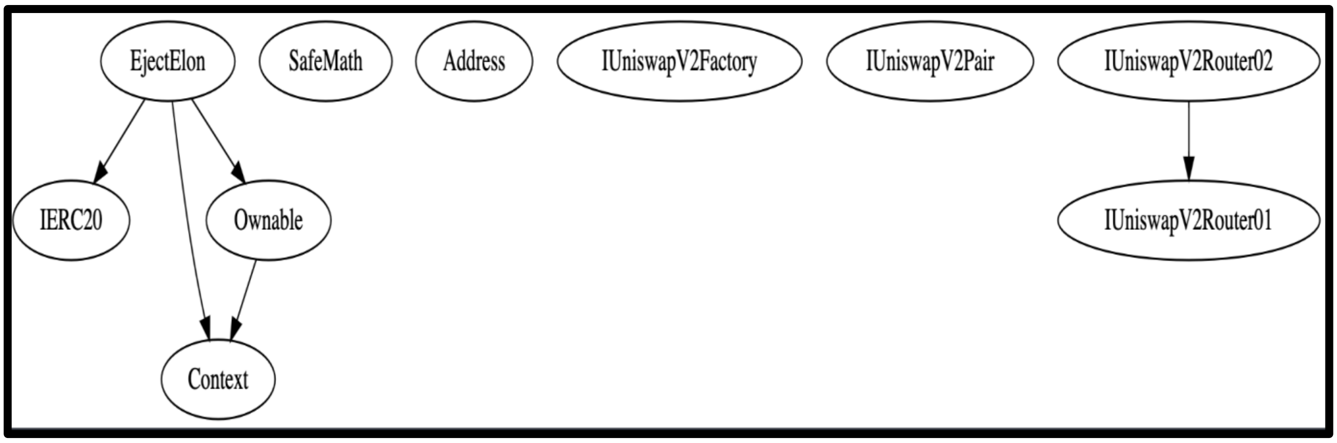
| | | | | |
|---|---|---|---|---|
| └ | setTaxFeePercent | External ❗️ | 🛑 | onlyOwner |
| └ | setLiquidityFeePercent | External ❗️ | 🛑 | onlyOwner |
| └ | setMaxTxPercent | External ❗️ | 🛑 | onlyOwner |
| └ | setSwapAndLiquifyEnabled | Public ❗️ | 🛑 | onlyOwner |
| └ | | External ❗️ | 💵 | NO❗️ |
| └ | _reflectFee | Private 🔐 | 🛑 | |
| └ | _getValues | Private 🔐 | | |
| └ | _getTValues | Private 🔐 | | |
| └ | _getRValues | Private 🔐 | | |
| └ | _getRate | Private 🔐 | | |
| └ | _getCurrentSupply | Private 🔐 | | |
| └ | _takeLiquidity | Private 🔐 | 🛑 | |
| └ | calculateTaxFee | Private 🔐 | | |
| └ | calculateLiquidityFee | Private 🔐 | | |
| └ | removeAllFee | Private 🔐 | 🛑 | |
| └ | restoreAllFee | Private 🔐 | 🛑 | |
| └ | isExcludedFromFee | Public ❗️ | | NO❗️ |
| └ | _approve | Private 🔐 | 🛑 | |
| └ | _transfer | Private 🔐 | 🛑 | |
| └ | swapAndLiquify | Private 🔐 | 🛑 | lockTheSwap |
| └ | swapTokensForEth | Private 🔐 | 🛑 | |
| └ | addLiquidity | Private 🔐 | 🛑 | |
| └ | _tokenTransfer | Private 🔐 | 🛑 | |

| | | | | |
|---|---|---|---|---|
| L | _transferStandard | Private 🔐 | 🛑 | |
| L | _transferToExcluded | Private 🔐 | 🛑 | |
| L | _transferFromExcluded | Private 🔐 | 🛑 | |

***Legend***

| **Symbol** | **Meaning** |
|---|---|
| 🛑 | **Function can modify state** |
| 💵 | **Function is payable** |

## Inheritance Hierarchy



## Security issue checking status

❖ **High severity issues**
No high severity issues found

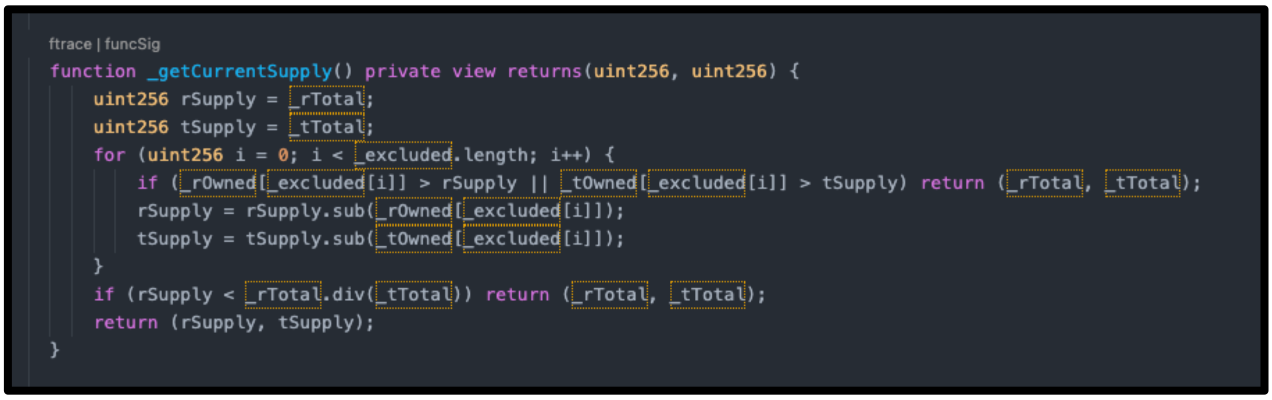
❖ **Medium severity issues**
No medium severity issues found

❖ **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.



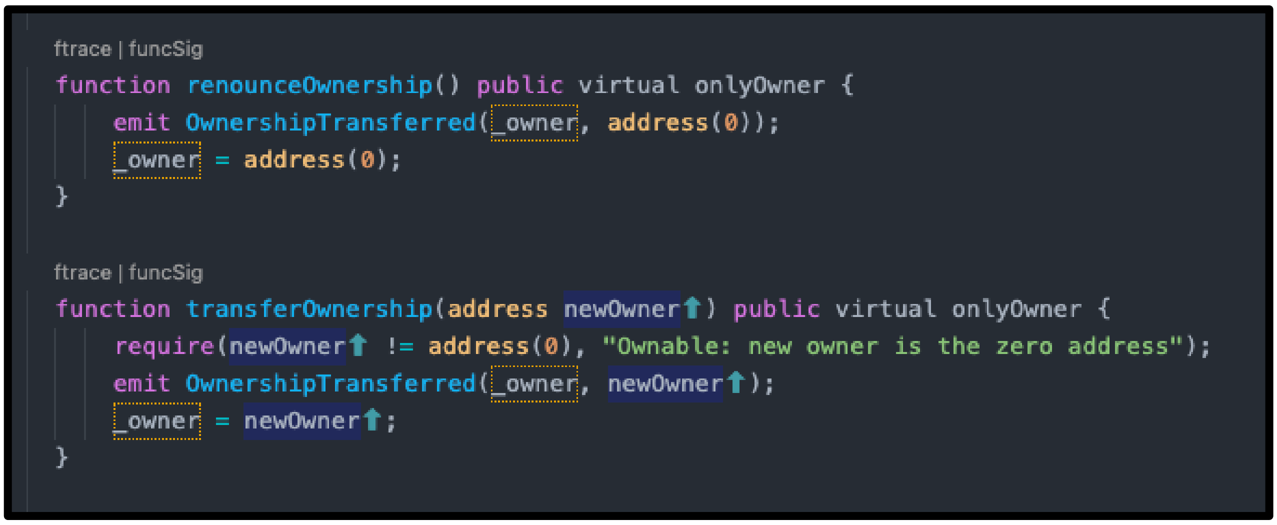
```
ftrace | funcSig
function _getCurrentSupply() private view returns(uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _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.

# Owner privileges

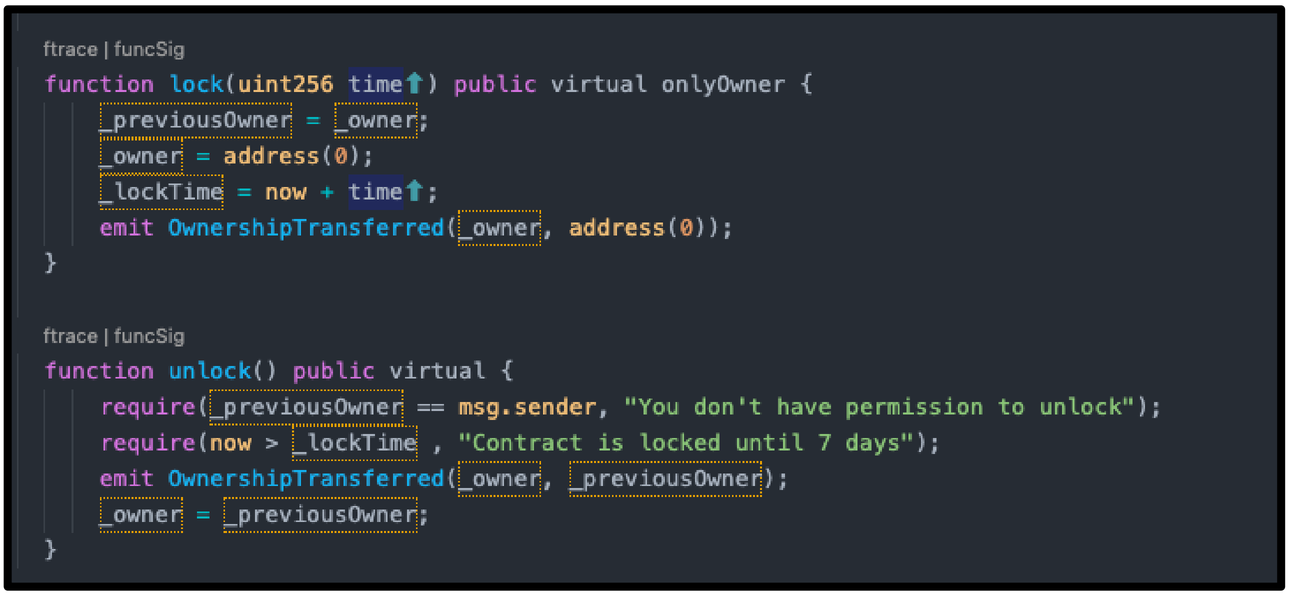
## (In the period when the owner is not renounced)

❖ Owner can transfer the contract and renounce the ownership.



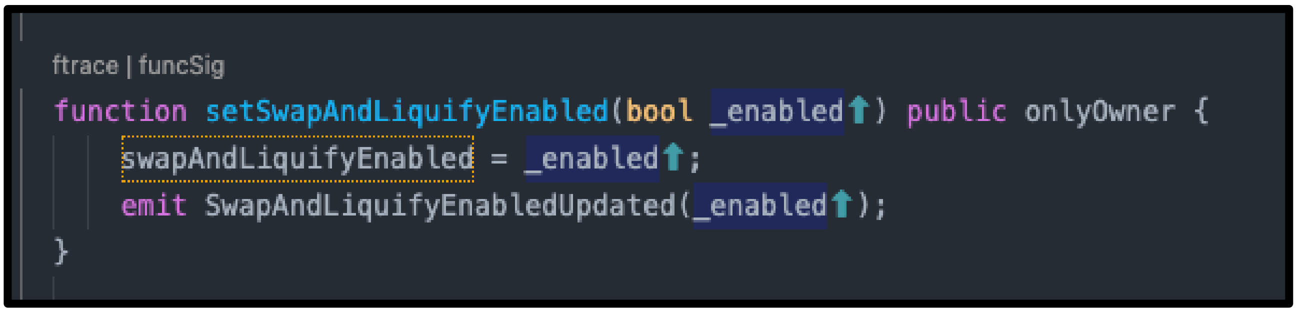
```
ftrace | funcSig
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}

ftrace | funcSig
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

❖ The contract will lock for the owner after transferring the ownership by specifying a period, and the contract will get unlocked once the specified period is over.



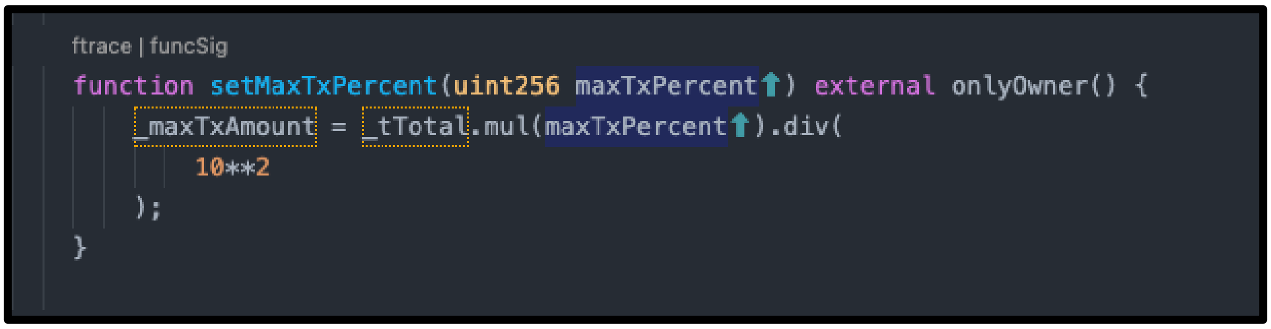
```
ftrace | funcSig
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = now + time;
    emit OwnershipTransferred(_owner, address(0));
}

ftrace | funcSig
function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(now > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

❖ Owner can swap and liquify (owner can enable and disable this).



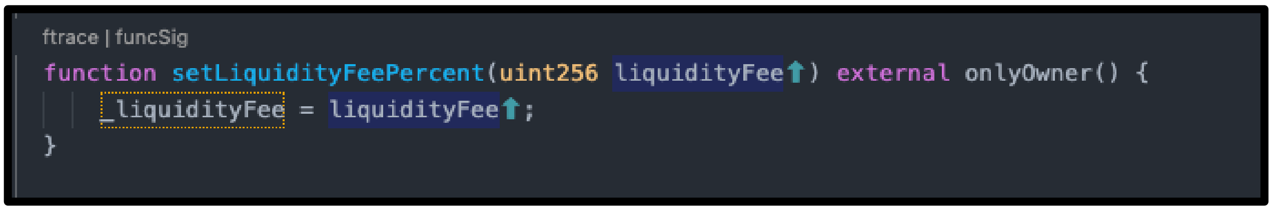
```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

❖ Owner can change maximum transaction amount.



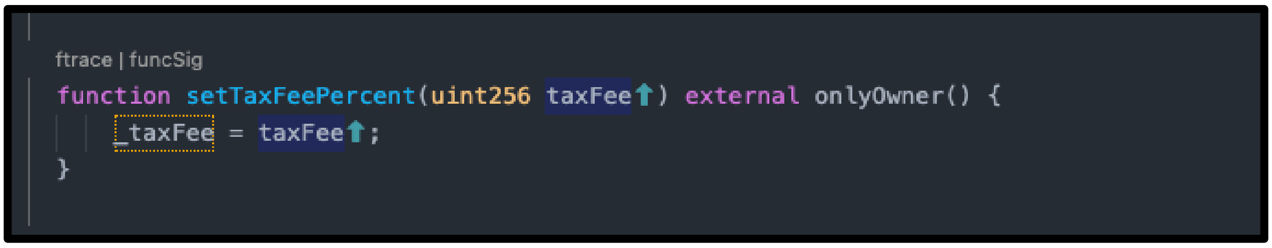
```
ftrace | funcSig
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**2
    );
}
```

❖ Owner can change Liquidity Fee percentage.



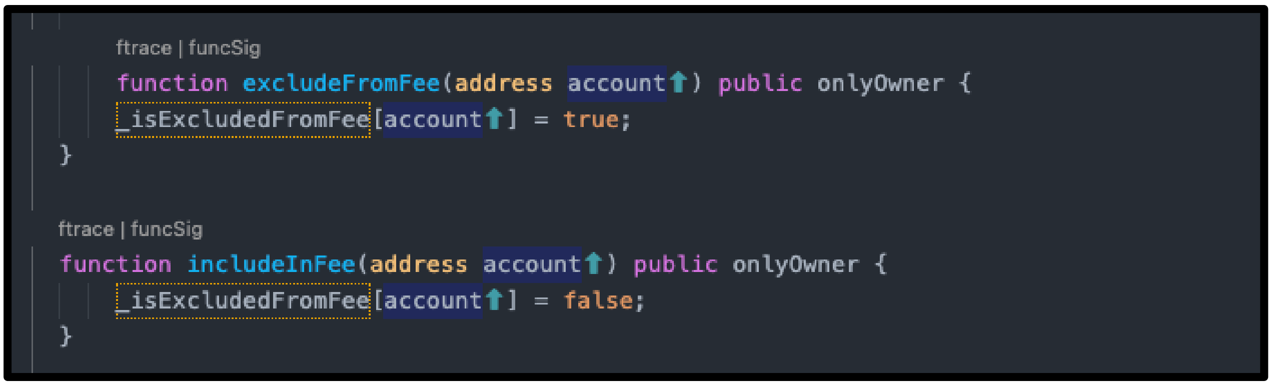
```
ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}
```

❖ Owner can change tax fee percentage



```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}
```

❖ Owner can Include and Exclude accounts from Fees.



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

ftrace | funcSig
function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

# Audit conclusion

While conducting the audit of the EJECT smart contract, it was observed that there is nothing alarming with the code and the contract contains only low severity issues.