# Smart Contract Security Audit

## Audit details:

| | |
|---|---|
| Audited project: | JaguarSwap |
| Deployer address | 0x3b94ef11edc19e7546711a7038a73ca9d16c416a |
| Blockchain: | Binance Smart Chain |
| Project website: | Not provided |

May, 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 JaguarSwap to perform an audit of smart contracts:

- https://bscscan.com/address/0x4a3524936Db5C310d852266033589D3f6F30BA5d#code
- https://bscscan.com/address/0x8e4301509A484c6fC211C8902013e90cD416F58D#code
- https://bscscan.com/address/0x402D745c21a792DAe1De4d38594F3b084d049B10#code
- https://bscscan.com/address/0x43EE4A63720b3D114638aE7678aC405f8Fafd578#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 01.05.2021.

| | |
|---|---|
| Contract name: | JaguarSwap |
| Compiler version: | v0.6.12+commit.27d51765 |
| Contract address: | 0x4a3524936Db5C310d852266033589D3f6F30BA5d |
| Total supply: | 250_000_000_000_000_000_000 |
| Token ticker: | JAGUAR |
| Decimals: | 18 |
| Token holders: | 180 |
| Transactions count: | 2191 |
| Top 100 holders dominance: | 99.11 % |
| Contract deployer address: | 0x3b94ef11edc19e7546711a7038a73ca9d16c416a |
| Contract's current owner address: | 0x8e4301509a484c6fc211c8902013e90cd416f58d |

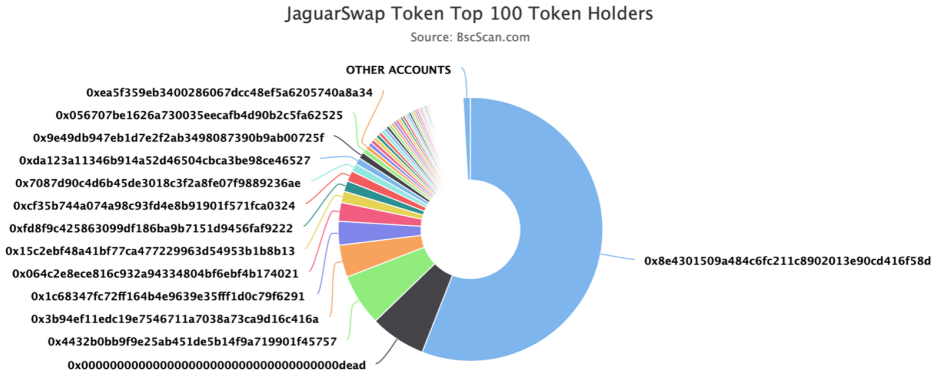
## JaguarSwap top 5 token holders

| Rank | Address | Quantity (Token) | Percentage |
|---|---|---|---|
| 1 | 0x8e4301509a484c6fc211c8902013e90cd416f58d | 139.87575190105347446 9 | 55.9503% |
| 2 | 0x000000000000000000000000000000000000dead | 17.055731125640490165 | 6.8223% |
| 3 | 0x4432b0bb9f9e25ab451de5b14f9a719901f45757 | 15.85 | 6.3400% |
| 4 | 0x3b94ef11edc19e7546711a7038a73ca9d16c416a | 9.937689607076073403 | 3.9751% |
| 5 | 0x1c68347fc72ff164b4e9639e35fff1d0c79f6291 | 7.254449971018335983 | 2.9018% |

# JaguarSwap top 100 token distribution

The top 100 holders collectively own 99.11% (247.77 Tokens) of JaguarSwap Token

Token Total Supply: 250.00 Token | Total Token Holders: 180

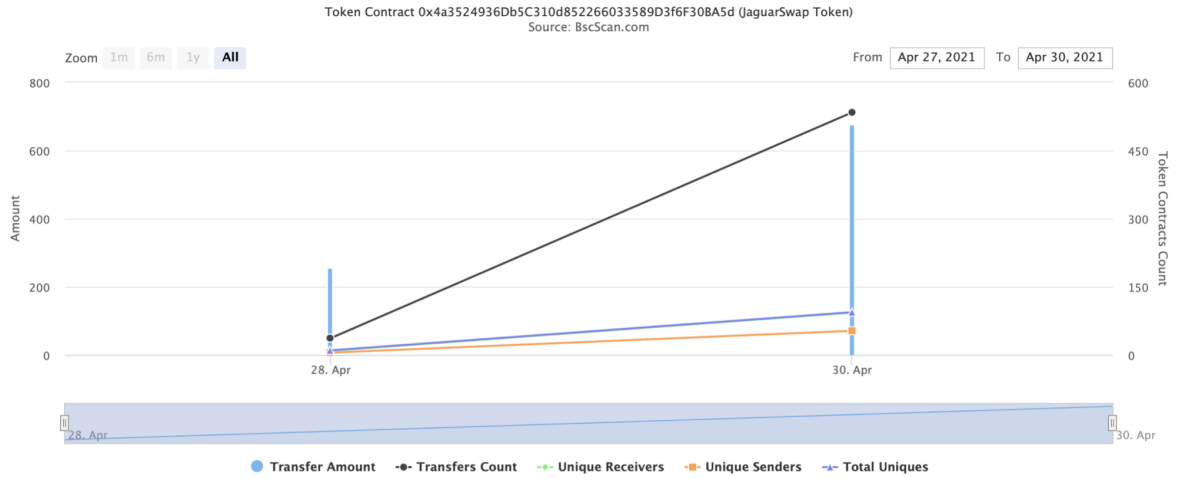



(A total of 247.77 tokens held by the top 100 accounts from the total supply of 250.00 token)

# JaguarSwap contract interaction details

Time Series: Token Contract Overview

Wed 28, Apr 2021 - Fri 30, Apr 2021



## Masterchef contract details for 01.05.2021.

| | |
|---|---|
| Contract name: | MasterChef |
| Compiler version: | v0.6.12+commit.27d51765 |
| Contract address: | 0x8e4301509A484c6fC211C8902013e90cD416F58D |
| Dev address: | 0x976aefa8878aa28b6dd52f789964c84a445e85d2 |
| Fee address: | 0xdbec8165bc99ca14c54281029a2505551fc5940a |
| Token contract address: | 0x4a3524936db5c310d852266033589d3f6f30ba5d |
| Token per block: | 1_000_000_000_000_000_000 |
| Contract owner address: | 0x402d745c21a792dae1de4d38594f3b084d049b10 |
| Pool length: | 27 |
| Start block: | 7109999 |
| Total alloc point: | 13400 |
| Bonus multiplier: | 1 |
| Referral commission rate: | 200 |
| Referral contract address: | 0x43ee4a63720b3d114638ae7678ac405f8fafd578 |

## MasterChef contract Pools info:

### Pool with id 0:

lpToken *address* : 0xfD8f9C425863099Df186BA9B7151d9456faf9222
allocPoint *uint256* : 4000
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 0

### Pool with id 1:

lpToken *address* : 0x1C68347FC72FF164b4E9639E35fFF1D0C79f6291
allocPoint *uint256* : 2400
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 0

### Pool with id 2:

lpToken *address* : 0x1B96B92314C44b159149f7E0303511fB2Fc4774f
allocPoint *uint256* : 500
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

### Pool with id 3:

lpToken *address* : 0xc15fa3E22c912A276550F3E5FE3b0Deb87B55aCd
allocPoint *uint256* : 400
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

### Pool with id 4:

lpToken *address* : 0x7561EEe90e24F3b348E1087A005F78B4c8453524
allocPoint *uint256* : 600
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

### Pool with id 5:

lpToken *address* : 0x70D8929d04b60Af4fb9B58713eBcf18765aDE422
allocPoint *uint256* : 600

lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 6:**

lpToken *address* : 0x3aB77e40340AB084c3e23Be8e5A6f7afed9D41DC
allocPoint *uint256* : 400
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 7:**

lpToken *address* : 0x680Dd100E4b394Bda26A59dD5c119A391e747d18
allocPoint *uint256* : 400
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 8:**

lpToken *address* : 0xbCD62661A6b1DEd703585d3aF7d7649Ef4dcDB5c
allocPoint *uint256* : 600
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 9:**

lpToken *address* : 0x0Ed8E0A2D99643e1e65CCA22Ed4424090B8B7458
allocPoint *uint256* : 200
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 10:**

lpToken *address* : 0xA527a61703D82139F8a06Bc30097cC9CAA2df5A6
allocPoint *uint256* : 200
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 11:**

lpToken *address* : 0x4a3524936Db5C310d852266033589D3f6F30BA5d
allocPoint *uint256* : 1000
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 0

**Pool with id 12:**

lpToken *address* : 0xe9e7CEA3DedcA5984780Bafc599bD69ADd087D56
allocPoint *uint256* : 200
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 13:**

lpToken *address* : 0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c
allocPoint *uint256* : 300
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 14:**

lpToken *address* : 0x55d398326f99059fF775485246999027B3197955
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 15:**

lpToken *address* : 0x7130d2A12B9BCbFAe4f2634d864A1Ee1Ce3Ead9c
allocPoint *uint256* : 200
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 16:**

lpToken *address* : 0x2170Ed0880ac9A755fd29B2688956BD959F933F8
allocPoint *uint256* : 200
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 17:**

lpToken *address* : 0x1AF3F329e8BE154074D8769D1FFa4eE058B1DBc3
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 18:**

lpToken *address* : 0x8AC76a51cc950d9822D68b83fE1Ad97B32Cd580d
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 19:**

lpToken *address* : 0x7083609fCE4d1d8Dc0C979AAb8c869Ea2C873402
allocPoint *uint256* : 200
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 20:**

lpToken *address* : 0x0E09FaBB73Bd3Ade0a17ECC321fD13a19e81cE82
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 21:**

lpToken *address* : 0x5Ac52EE5b2a633895292Ff6d8A89bB9190451587
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 22:**

lpToken *address* : 0xa184088a740c695E156F91f5cC086a06bb78b827
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999

accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 23:**

lpToken *address* : 0xF952Fc3ca7325Cc27D15885d37117676d25BfdA6
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 24:**

lpToken *address* : 0xC9849E6fdB743d08fAeE3E34dd2D1bc69EA11a51
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 25:**

lpToken *address* : 0xBf5140A22578168FD562DCcF235E5D43A02ce9B1
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

**Pool with id 26:**

lpToken *address* : 0xCa3F508B8e4Dd382eE878A314789373D80A5190A
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 7109999
accJaguarPerShare *uint256* : 0
depositFeeBP *uint16* : 400

# 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. | Medium issues |
| 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

### 1. Wrong burning

Issue:

There is sending tokens to the dead address in overridden _transfer functions, instead of burning them in token contract.

```
function _transfer(address sender, address recipient, uint256 amount) internal virtual override {
    if (recipient == BURN_ADDRESS) {
        super._transfer(sender, recipient, amount);
    } else {
        // 2% of every transfer burnt
        uint256 burnAmount = amount.mul(2).div(100);
        // 98% of transfer sent to recipient
        uint256 sendAmount = amount.sub(burnAmount);
        require(amount == sendAmount + burnAmount, "JAGUAR::transfer: Burn value invalid");

        super._transfer(sender, BURN_ADDRESS, burnAmount);
        super._transfer(sender, recipient, sendAmount);
        amount = sendAmount;
    }
}
```

Recommendation:

There should be a burn instead of sending to the dead address.

## Low Severity Issues

### 1. Block gas limit

Issue:

The updateEmissionRate function can fail due to block gas limit if the pool size is too big.

```
function updateEmissionRate() public {
    require(block.number > startBlock, "updateEmissionRate: Can only be called after mining starts");
    require(jaguarPerBlock > MINIMUM_EMISSION_RATE, "updateEmissionRate: Emission rate has reached the minimum threshold");

    uint256 currentIndex = block.number.sub(startBlock).div(EMISSION_REDUCTION_PERIOD_BLOCKS);
    if (currentIndex <= lastReductionPeriodIndex) {
        return;
    }

    uint256 newEmissionRate = jaguarPerBlock;
    for (uint256 index = lastReductionPeriodIndex; index < currentIndex; ++index) {
        newEmissionRate = newEmissionRate.mul(1e4 - EMISSION_REDUCTION_RATE_PER_PERIOD).div(1e4);
    }

    newEmissionRate = newEmissionRate < MINIMUM_EMISSION_RATE ? MINIMUM_EMISSION_RATE : newEmissionRate;
    if (newEmissionRate >= jaguarPerBlock) {
        return;
    }

    massUpdatePools();
    lastReductionPeriodIndex = currentIndex;
    uint256 previousEmissionRate = jaguarPerBlock;
    jaguarPerBlock = newEmissionRate;
    emit EmissionRateUpdated(msg.sender, previousEmissionRate, newEmissionRate);
}
```

## 2. add function issue

Issue:

If some LP token is added to the contract twice using function add, then the total amount of reward jaguarReward in function updatePool will be incorrect.

```
function add(uint256 _allocPoint, IBEP20 _lpToken, uint16 _depositFeeBP, bool _withUpdate) public onlyOwner {
    require(_depositFeeBP <= 10000, "add: invalid deposit fee basis points");
    if (_withUpdate) {
        massUpdatePools();
    }
    uint256 lastRewardBlock = block.number > startBlock ? block.number : startBlock;
    totalAllocPoint = totalAllocPoint.add(_allocPoint);
    poolInfo.push(PoolInfo({
        lpToken: _lpToken,
        allocPoint: _allocPoint,
        lastRewardBlock: lastRewardBlock,
        accJaguarPerShare: 0,
        depositFeeBP: _depositFeeBP
    }));
}
```

Recommendation:

Add the mapping from address to bool and check that same address will not be added twice.

# Owner privileges

- Owner can withdraw tokens sent by mistake from the Referral contract.
- Owner can change the operator of the Referral contract and record Referral.
- Owner can change the jaguar referral.

# Conclusion

Smart contracts contain medium severity and low 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.*