RUGFREECOINS

# ZANDREUM Token

RugfreeCoins Verified on February 29th, 2024

# Overview

✅ No mint function found, the owner cannot mint tokens after initial deployment.

✅ The owner can't set a max transaction limit

✅ The owner can't pause trading once it's enabled

❌ The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

✅ The owner can't change fees.

✅ The owner can't blacklist wallets.

✅ The owner can't set a max wallet limit

❌ The owner can claim the contract's balance of its own token.

## ❗ HIGH SEVERITY ISSUES

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

```
function enableTrading() public onlyOwner{
    require(!tradingEnabled, "Trading is already enabled");
    tradingEnabled = true;
}
```

Anyone can call the errorToken function and send native tokens from the token contract to the developer's wallet.

```
function errorToken(address _token) external {
  ERC20(_token).transfer(DEVAddress, IERC20(_token).balanceOf(address(this)));
}
```

The owner can claim native tokens from the contract

```
function claimStuckTokens(address token) external onlyOwner {
    if (token == address(0x0)) {
        (bool success,) = msg.sender.call{value: address(this).balance}("");
        require(success, "Claim failed");
        return;
    }
    IERC20 ERC20token = IERC20(token);
    uint256 balance = ERC20token.balanceOf(address(this));
    ERC20token.transfer(msg.sender, balance);
}
```

The owner has the ability to change the swap threshold without any maximum limit. If the owner sets this to a very large amount, swaps may fail when the contract attempts to sell an excessively large amount of tokens at once, consequently causing the sales to also fail.

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
    require(newAmount > totalSupply() / 1000000, "SwapTokensAtAmount must be
greater than 0.0001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}
```

# Contents

# Audit details

**Audited project**

ZANDREUM Token

**Contract Address**

0x11FF100785f0AF075a96D2f8c7ec710913abD4EC

**Client contact**

ZANDREUM Token Team

**Blockchain**

Binance Smart chain

**Project website**

https://zandreum.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 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 the **ZANDREUM Token Team** to perform an audit of the smart contract.

**https://bscscan.com/token/0x11FF100785f0AF075a96D2f8c7ec710913abD4EC**

This audit focuses on verifying 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, and long-term sustainability, and as a guide to improving the smart contract's security posture by remediating the identified issues.

# Tokenomics

**▲ 0% tax when buying**

**▲ 0% tax when selling (can change up to 10%)**

10% of trade goes to the Dev wallet in BNB

---

# Target market and the concept

- Anyone who's interested in the Crypto space with long-term investment plans.
- Anyone who's ready to earn a passive income by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in taking part in the ZANDREUM token ecosystem.
- Anyone who's interested in taking part in the future plans of ZANDREUM Token.
- Anyone who's interested in making financial transactions with any other party ZANDREUM Token as the currency.

# Potential to grow with score points

| | |
|---|---|
| ⚡ Project efficiency | 8 / 10 |
| 🌟 Project uniqueness | 8 / 10 |
| 📊 Information quality | 8 / 10 |
| 👌 Service quality | 8 / 10 |
| 💻 System quality | 8 / 10 |
| 🌍 Impact on the community | 8 / 10 |
| 💼 Impact on the business | 9 / 10 |
| 🔮 Preparing for the future | 8 / 10 |
| 🔒 Smart contract security | 6 / 10 |
| 🛠️ Smart contract functionality assessment | 9 / 10 |
| 🏆 **Total Score** | **8.0** / **10** |

# Contract details

Token contract details for 29th of February 2024

| | |
|---|---|
| Contract name | **ZANDREUM** |
| Contract address | **0x11FF100785f0AF075a96D2f8c7ec710913abD4EC** |
| Token supply | **10,000,000,000** |
| Token ticker | **ZANDREUM** |
| Decimals | **18** |
| Token holders | **3** |
| Transaction count | **3** |
| Contract deployer address | **0x9f5bA2CBe187f6d21dfb3285571aF5620A7CCb6b** |
| Contract’s current owner address | **0x9f5bA2CBe187f6d21dfb3285571aF5620A7CCb6b** |

# Contract code function details

| № | Category | Item | Result |
|---|---|---|---|
| 1 | Coding conventions | ERC20 Token standards | PASS |
| | | Compile errors | PASS |
| | | Compiler version security | PASS |
| | | Visibility specifiers | PASS |
| | | Gas consumption | PASS |
| | | 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 |
| | | Self destruct function security | PASS |
| 3 | Business security & centralisation | Access control of owners | HIGH |
| | | Business logics | PASS |
| | | Business implementation | LOW |
| 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 |

# Contract description table

The 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 their visibility and mutability.

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| **IPancakeswapV2 Factory** | **Interface** | | | |
| L | feeTo | External ❗ | | NO ❗ |
| L | feeToSetter | External ❗ | | NO ❗ |
| L | getPair | External ❗ | | NO ❗ |
| L | allPairs | External ❗ | | NO ❗ |
| L | allPairsLength | External ❗ | | NO ❗ |
| L | createPair | External ❗ | 🛑 | NO ❗ |
| L | setFeeTo | External ❗ | 🛑 | NO ❗ |
| L | setFeeToSetter | External ❗ | 🛑 | NO ❗ |
| | | | | |
| **IPancakeswapV2 Router01** | **Interface** | | | |
| L | factory | External ❗ | | NO ❗ |
| L | WETH | External ❗ | | NO ❗ |
| L | addLiquidity | External ❗ | 🛑 | NO ❗ |
| L | addLiquidityETH | External ❗ | 💵 | NO ❗ |
| L | removeLiquidity | External ❗ | 🛑 | NO ❗ |
| L | removeLiquidityETH | External ❗ | 🛑 | NO ❗ |
| L | removeLiquidityWithPermit | External ❗ | 🛑 | NO ❗ |
| L | removeLiquidityETHWithPermit | External ❗ | 🛑 | NO ❗ |
| L | swapExactTokensForTokens | External ❗ | 🛑 | NO ❗ |

| | | | | |
|---|---|---|---|---|
| L | 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 ❗ |
| | | | | |
| **IPancakeswapV2 Router02** | **Interface** | **IPancakeswap V2 Router01** | | |
| L | removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO ❗ |
| L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO ❗ |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO ❗ |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 | NO ❗ |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO ❗ |
| | | | | |
| **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 ❗ |
| | | | | |

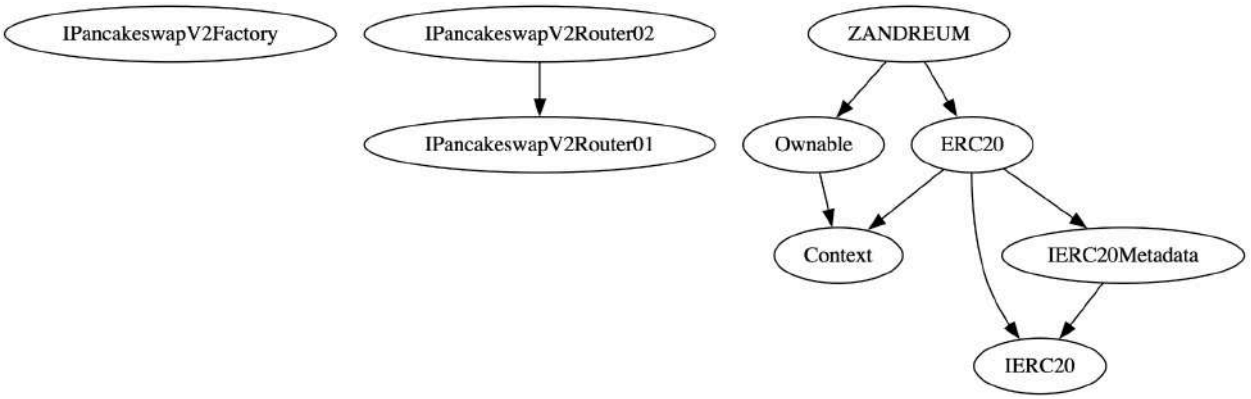
| IERC20Metadata | Interface | IERC20 | | |
|---|---|---|---|---|
| L | name | External ❗ | | NO ❗ |
| L | symbol | External ❗ | | NO ❗ |
| L | decimals | External ❗ | | NO ❗ |
| | | | | |
| **Context** | **Implementation** | | | |
| L | _msgSender | Internal 🔒 | | |
| L | _msgData | Internal 🔒 | | |
| | | | | |
| **Ownable** | **Implementation** | **Context** | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | owner | Public ❗ | | NO ❗ |
| L | renounceOwnership | Public ❗ | 🛑 | onlyOwner |
| L | transferOwnership | Public ❗ | 🛑 | onlyOwner |
| | | | | |
| **ERC20** | **Implementation** | **Context, IERC20, IERC20 Metadata** | | |
| 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 | _transfer | Internal 🔒 | 🛑 | |
| L | _init | Internal 🔒 | 🛑 | |
| L | _burn | Internal 🔒 | 🛑 | |
| L | _approve | Internal 🔒 | 🛑 | |
| L | _beforeTokenTransfer | Internal 🔒 | 🛑 | |
| L | _afterTokenTransfer | Internal 🔒 | 🛑 | |
| | | | | |
| **ZANDREUM** | **Implementation** | **ERC20, Ownable** | | |
| L | | Public ❗ | 🛑 | ERC20 |
| L | | External ❗ | 💵 | NO ❗ |
| L | enableTrading | Public ❗ | 🛑 | onlyOwner |
| L | errorBalance | External ❗ | 🛑 | NO ❗ |
| L | errorToken | External ❗ | 🛑 | NO ❗ |
| L | claimStuckTokens | External ❗ | 🛑 | onlyOwner |
| L | excludeFromFees | External ❗ | 🛑 | onlyOwner |
| L | isExcludedFromFees | Public ❗ | | NO ❗ |
| L | setSellFee | External ❗ | 🛑 | onlyOwner |
| L | _transfer | Internal 🔒 | 🛑 | |
| L | setSwapTokensAtAmount | External ❗ | 🛑 | onlyOwner |
| L | setSwapWithLimit | External ❗ | 🛑 | onlyOwner |
| L | swap | Private 🔐 | 🛑 | |

## Legend

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

# Inheritance Hierarchy



# Security issue checking status

❖ High severity issues

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

```
function enableTrading() public onlyOwner{
    require(!tradingEnabled, "Trading is already enabled");
    tradingEnabled = true;
}
```

Anyone can call the errorToken function and send native tokens from the token contract to the developer's wallet.

```
function errorToken(address _token) external {
  ERC20(_token).transfer(DEVAddress, IERC20(_token).balanceOf(address(this)));
}
```

The owner can claim native tokens from the contract

```
function claimStuckTokens(address token) external onlyOwner {
    if (token == address(0x0)) {
        (bool success,) = msg.sender.call{value: address(this).balance}("");
        require(success, "Claim failed");
        return;
    }
    IERC20 ERC20token = IERC20(token);
    uint256 balance = ERC20token.balanceOf(address(this));
    ERC20token.transfer(msg.sender, balance);
}
```

The owner has the ability to change the swap threshold without any maximum limit. If the owner sets this to a very large amount, swaps may fail when the contract attempts to sell an excessively large amount of tokens at once, consequently causing the sales to also fail.

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
    require(newAmount > totalSupply() / 1000000, "SwapTokensAtAmount must be
greater than 0.0001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}
```

- Medium severity issues

  No medium severity issues found

- Low severity issues

The buy fee and wallet-to-wallet fees are default-set to 0, and the owner has no option to change them later.

```
buyFee  = 0;
sellFee = 0;
walletToWalletTransferFee = 0;
```

# Owner privileges

- Owner can enable trading, once enabled can not disable again

```
function enableTrading() public onlyOwner{
    require(!tradingEnabled, "Trading is already enabled");
    tradingEnabled = true;
}
```

- Anyone can call this function and send BNB from contract to dev wallet

```
function errorBalance() external {
  payable(DEVAddress).transfer(address(this).balance);
}
```

- Anyone call this function and send any BEP20 tokens from the contract (including native tokens) to dev wallet

```
function errorToken(address _token) external {
  ERC20(_token).transfer(DEVAddress, IERC20(_token).balanceOf(address(this)));
}
```

❖ Owner can claim any BEP20 tokens from the contract (including native tokens)

```
function claimStuckTokens(address token) external onlyOwner {
    if (token == address(0x0)) {
        (bool success,) = msg.sender.call{value: address(this).balance}("");
        require(success, "Claim failed");
        return;
    }
    IERC20 ERC20token = IERC20(token);
    uint256 balance = ERC20token.balanceOf(address(this));
    ERC20token.transfer(msg.sender, balance);
}
```

❖ Owner can include/exclude wallets from fees

```
function excludeFromFees(address account, bool excluded) external onlyOwner{
    require(_isExcludedFromFees[account] != excluded,"Account is already the
value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}
```

❖ Owner can change sell fees maximum up to 10%

```
function setSellFee(uint256 _sellFee) external onlyOwner {
    require(_sellFee <= 10, "Sell Fee cannot be more than 10%");
    sellFee = _sellFee;
    emit SellFeeUpdated(sellFee);
}
```

- Owner can change swap point without max limit

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
    require(newAmount > totalSupply() / 1000000, "SwapTokensAtAmount must be
greater than 0.0001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}
```

- Owner can enable/disable swap with limit, if enabled maximum swap tokens amount per swap will be set to swap threshold

```
function setSwapWithLimit(bool _swapWithLimit) external onlyOwner{
    swapWithLimit = _swapWithLimit;
    emit SwapWithLimitUpdated(swapWithLimit);
}
```

# Audit conclusion

RugFreeCoins team has performed in-depth testing, line-by-line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

| | |
|---|---|
| Smart contract functional Status: | PASS |
| Smart contract security Status: | HIGH ISSUES & LOW ISSUE |
| Number of risk issues: | 05 |
| Solidity code functional issue level: | PASS |
| Number of owner privileges: | 08 |
| Centralization risk correlated to the active owner: | HIGH |
| Smart contract active ownership: | ACTIVE |