



# Smart Contract Audit

FOR

## Rocket Rush

DATED : 02 Mar 23'



# AUDIT SUMMARY

---

**Project name** – Rocket Rush

**Date:** 02 March, 2023

**Scope of Audit-** Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

**Audit Status:** Passed (Contract is developed by pinksale's Safu Dev)

## Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

---

# USED TOOLS

---

## Tools:

### 1- Manual Review:

a line by line code review has been performed by audit ace team.

### 2- BSC Test Network:

all tests were done on BSC Test network, each test has its transaction has attached to it.

### 3- Slither : Static Analysis

**Testnet Link:** all tests were done using this contract, tests are done on BSC Testnet

<https://testnet.bscscan.com/token/0xeEcb2762098A7Ef190b4BC821730230753ef6e10#code>

---



# Token Information

---

**Token Name :** Rocket Rush

**Token Symbol:** RR

**Decimals:** 18

**Token Supply:** 100,000,000

**Token Address:** Not provided

**Checksum:**

cb8957a126d502f6f5c9e873a5ecf32e8941eb1c

---



# TOKEN OVERVIEW

---

## **Fees:**

Buy Fees: 1%

Sell Fees: 3%

Transfer Fees: 0%

---

**Fees Privilege:** Owner

---

**Ownership :** Owned

---

**Minting:** No mint function

---

**Max Tx Amount/ Max Wallet Amount:** No

---

**Blacklist:** No

---

**Other Privileges:** including and excluding from fees

---



# AUDIT METHODOLOGY

---

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
  - Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
  - Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
  - Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
  - Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
  - Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.
-

# VULNERABILITY CHECKLIST

---

- |  |   |
|--|---|
|  Return values of low-level calls  |  Gasless Send                  |
|  Private modifier                  |  Using block.timestamp         |
|  Multiple Sends                    |  Re-entrancy                   |
|  Using Suicide                    |  Tautology or contradiction   |
|  Gas Limitand Loops              |  Timestamp Dependence        |
|  Address hardcoded               |  Revert/require functions    |
|  Exception Disorder              |  Use of tx.origin            |
|  Using inline assembly           |  Integer overflow/underflow  |
|  Divide before multiply          |  Dangerous strict equalities |
|  Missing Zero Address Validation |  Using SHA3                  |
|  Compiler version not fixed      |  Using throw                 |
-



# CLASSIFICATION OF RISK

## Severity

## Description

◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization / Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

## Findings

### Severity

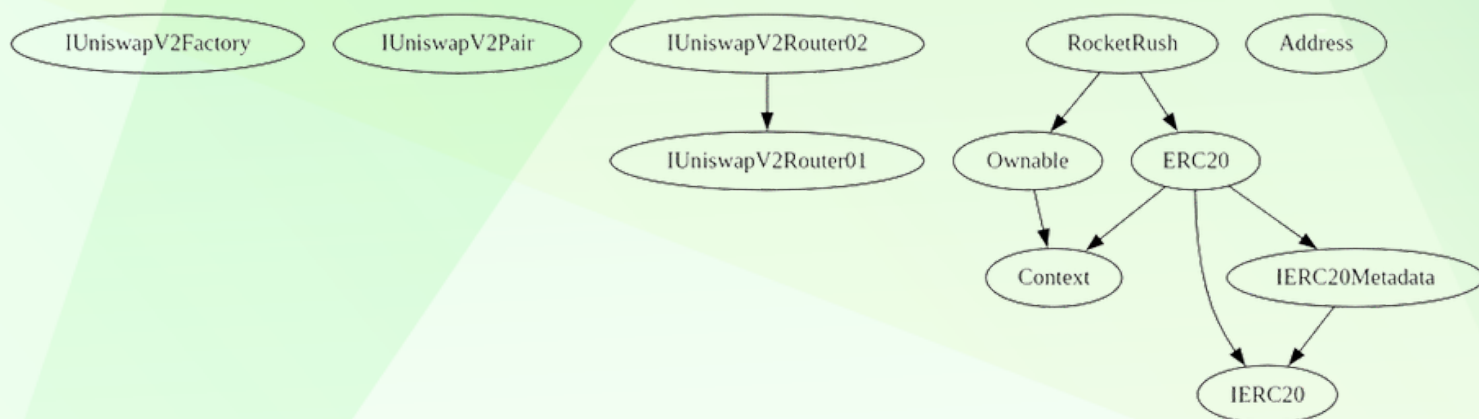
### Found

◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0



# INHERITANCE TREE

---



# POINTS TO NOTE

---

- **sum of buy and sell taxes can not be more than 7%**
  - **Owner is not able to set transfer taxes (0% static)**
  - **Owner is not able to set max wallet/transfer/buy/sell amounts**
  - **Owner is not able to blacklist an arbitrary wallet**
  - **Owner is not able to disable trades**
  - **Owner is not able to mint new tokens**
-



# CONTRACT ASSESMENT

Contract	Type	Bases			
└──┐	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
**IUniswapV2Factory**	Interface				
└ 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	!



# CONTRACT ASSESMENT


|||||


| **\*\*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! |

| | swapExactETHForTokens | External ! |  | NO! |

| | swapTokensForExactETH | External ! |  | NO! |

| | swapExactTokensForETH | External ! |  | NO! |

| | swapETHForExactTokens | External ! |  | NO! |

| | quote | External ! | | NO! |

| | getAmountOut | External ! | | NO! |

| | getAmountIn | External ! | | NO! |

| | getAmountsOut | External ! | | NO! |


| | getAmountsIn | External ! | | NO! |


|||||

| **\*\*IUniswapV2Router02\*\*** | Interface | IUniswapV2Router01 |||

| | removeLiquidityETHSupportingFeeOnTransferTokens | External ! |  | NO! |

| | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! |  | NO! |

| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! |  | NO! |

| | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! |  | NO! |

| | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! |  | NO! |

|||||

| **\*\*IERC20\*\*** | Interface | |||

| | totalSupply | External ! | | NO! |

| | balanceOf | External ! | | NO! |

| | transfer | External ! |  | NO! |

| | allowance | External ! | | NO! |

| | approve | External ! |  | NO! |

| | transferFrom | External ! |  | NO! |

|||||

| **\*\*IERC20Metadata\*\*** | Interface | IERC20 |||

| | name | External ! | | NO! |

| | symbol | External ! | | NO! |

| | decimals | External ! | | NO! |

|||||










# CONTRACT ASSESMENT

```
| **Address** | Library | ||| |
| | sendValue | Internal | 🔒 | 🔴 | |
| |||||
| **Context** | Implementation | |||
| | _msgSender | Internal | 🔒 | | |
| | _msgData | Internal | 🔒 | | |
| |||||
| **Ownable** | Implementation | Context | |||
| | <Constructor> | Public | ! | 🔴 | NO! |
| | owner | Public | ! | | NO! |
| | renounceOwnership | Public | ! | 🔴 | onlyOwner |
| | transferOwnership | Public | ! | 🔴 | onlyOwner |
| |||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata | |||
| | <Constructor> | Public | ! | 🔴 | NO! |
| | name | Public | ! | | NO! |
| | symbol | Public | ! | | NO! |
| | decimals | Public | ! | | NO! |
| | totalSupply | Public | ! | | NO! |
| | balanceOf | Public | ! | | NO! |
| | transfer | Public | ! | 🔴 | NO! |
| | allowance | Public | ! | | NO! |
| | approve | Public | ! | 🔴 | NO! |
| | transferFrom | Public | ! | 🔴 | NO! |
| | increaseAllowance | Public | ! | 🔴 | NO! |
| | decreaseAllowance | Public | ! | 🔴 | NO! |
| | _transfer | Internal | 🔒 | 🔴 | |
| | _mint | Internal | 🔒 | 🔴 | |
| | _burn | Internal | 🔒 | 🔴 | |
| | _approve | Internal | 🔒 | 🔴 | |
| | _beforeTokenTransfer | Internal | 🔒 | 🔴 | |
| | _afterTokenTransfer | Internal | 🔒 | 🔴 | |
| |||||
| **RocketRush** | Implementation | ERC20, Ownable | |||
| | <Constructor> | Public | ! | 🔴 | ERC20 |
| | <Receive Ether> | External | ! | 🏠 | NO! |
| | claimStuckTokens | External | ! | 🔴 | onlyOwner |
| | excludeFromFees | External | ! | 🔴 | onlyOwner |
| | isExcludedFromFees | Public | ! | | NO! |
| | updateBuyFees | External | ! | 🔴 | onlyOwner |
| | updateSellFees | External | ! | 🔴 | onlyOwner |
| | changeMarketingWallet | External | ! | 🔴 | onlyOwner |
```




# CONTRACT ASSESMENT


---

<sup>L</sup>	changeEcosystemWallet	External !		onlyOwner
<sup>L</sup>	enableTrading	External !		onlyOwner
<sup>L</sup>	\_transfer	Internal 		
<sup>L</sup>	setSwapTokensAtAmount	External !		onlyOwner
<sup>L</sup>	swapAndSendBNB	Private 		

| Symbol | Meaning |

|:-----:|-----|

|  | Function can modify state |

|  | Function is payable |



# STATIC ANALYSIS

```
Reentrancy in RocketRush.swapAndSendBNB(uint256) (contracts/Token.sol#872-903):
  External calls:
    - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#879-885)
    - address(marketingWallet).sendValue(marketingBalance) (contracts/Token.sol#895)
    - address(ecosystemWallet).sendValue(ecosystemBalance) (contracts/Token.sol#899)
  Event emitted after the call(s):
    - SwapAndSendBNB(tokenAmount,newBalance) (contracts/Token.sol#902)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Context_msgData() (contracts/Token.sol#390-393) is never used and should be removed
ERC20_burn(address,uint256) (contracts/Token.sol#580-595) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.17 (contracts/Token.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.18 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#371-382):
  - (success) = recipient.call{value: amount}() (contracts/Token.sol#377)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#65) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#67) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#90) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#138) is not in mixedCase
Parameter RocketRush.updateBuyFees(uint256,uint256)._marketingFeeOnBuy (contracts/Token.sol#738) is not in mixedCase
Parameter RocketRush.updateBuyFees(uint256,uint256)._ecosystemFeeOnBuy (contracts/Token.sol#739) is not in mixedCase
Parameter RocketRush.updateSellFees(uint256,uint256)._marketingFeeOnSell (contracts/Token.sol#753) is not in mixedCase
Parameter RocketRush.updateSellFees(uint256,uint256)._ecosystemFeeOnSell (contracts/Token.sol#754) is not in mixedCase
Parameter RocketRush.changeMarketingWallet(address)._marketingWallet (contracts/Token.sol#768) is not in mixedCase
Parameter RocketRush.changeEcosystemWallet(address)._ecosystemWallet (contracts/Token.sol#784) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#391)" inContext (contracts/Token.sol#385-394)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#143) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#144)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

RocketRush.isBlacklisted (contracts/Token.sol#629) is never used in RocketRush (contracts/Token.sol#622-904)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

RocketRush.uniswapV2Pair (contracts/Token.sol#626) should be immutable
RocketRush.uniswapV2Router (contracts/Token.sol#625) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

**Result => A static analysis of contract's source code has been performed using slither,  
No major issues were found in the output**



# FUNCTIONAL TESTING

---

## **Router (PCS V2):**

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All functionalities of the token appear to be working as intended. We have examined the smart contract's code and implemented various test cases for:

- Buying (excluded and not excluded from fees)
  - Selling (excluded and not excluded from fees)
  - Transferring (excluded and not excluded from fees)
  - Internal Swap (if contract balance is more than threshold)
- our analysis did not reveal any potential issues or security flaws.

## **1- Adding Liquidity (Passed):**

### **liquidity added on Pancakeswap V2:**

<https://testnet.bscscan.com/tx/0x0f9eb169f9426843288d8e4632d99c30a07479b1fe89c9e4265b113a951b13db>

**no issue were found on adding liquidity.**

## **2- Buying when excluded (0% Tax) (Passed):**

<https://testnet.bscscan.com/tx/0x83cc0d75c82a814b32a6951e37293275161cb73ebbabbbfb98737794d3db95b71>

## **3- Selling when excluded (0% Tax) (Passed):**

<https://testnet.bscscan.com/tx/0xce284267615c72237db9a29e84590d9c1b0f3f9b5e9aa7cc7f4bcb2ea2668413>

---





# FUNCTIONAL TESTING

---

## 4-Transferring when excluded (0% tax )(Passed):

<https://testnet.bscscan.com/tx/0x789fa1f6e25221da03bd08a8ef7bd3b1a2d48237b1fea6fce045d0d10b7b1cc0>

## 5- Buying when not excluded (1% tax) (passed):

<https://testnet.bscscan.com/tx/0xb6e03040aaa392d0087068be7ae04ec772b5ba5f09ab7dce55e58436353f8d24>

## 6- Selling when not excluded (3% tax) (passed):

<https://testnet.bscscan.com/tx/0x5c6ba8c6542ac72614688b046a30432189327e6a14de6e6da5b6ba5753f1d83f>

## 7- Transferring when not excluded (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x52b801146ac25cc409db588036d8be80fb4115ef1d898fef375f556d383b8574>

## 8- Internal swap (passed):

### ecosystem wallet received ETH

<https://testnet.bscscan.com/address/0x1248c6aa5aec6f7baa7171b097e1baad657abaf7#internaltx>

### marketing wallet received ETH

<https://testnet.bscscan.com/address/0x24b416debb518401462afd8a0a84d1e7a804406e>

---



# MANUAL TESTING

---

**NO ISSUES FOUND**

A solid red vertical bar is located in the bottom right corner of the page.



# DISCLAIMER

---

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.

---



# ABOUT AUDITACE

---

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



**<https://auditace.tech/>**



**[https://t.me/Audit\\_Ace](https://t.me/Audit_Ace)**



**[https://twitter.com/auditace\\_](https://twitter.com/auditace_)**



**<https://github.com/Audit-Ace>**

---