SECURITY AUDIT OF

LEGEND OF RPS TOKEN

Public Report

Sep 08, 2022

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.

EXECUTIVE SUMMARY

This Security Audit Report prepared by Verichains Lab on Sep 08, 2022. We would like to thank the Legend of RPS for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Legend of RPS Token. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified no vulnerable issues in the smart contracts code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About Legend of RPS Token

Legend of RPS is a classical game which usually called Rock Paper Scissors - now available on blockchain NFT technology with a brand new superb level! In Rock Paper Scissors you'll win if you're lucky enough. But in Legend of RPS, there's no luck! You must be accurate 100% and able to take the super high pressure of time to tame this tough but addictive game! All your task in Legend of RPS is picking the right result for Rock Paper Scissors battle within tik tak time! You have to do it right and fast cause the timer is flashing! Legend of RPS rule is still the same as Rock Paper Scissors:

- Scissors cut paper.
- Paper covers rock.
- Rock crushes scissors.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of Legend of RPS Token.

The `LegendOfRPS` contract was deployed on Binance Smart Chain Mainnet at address `0x57f251706a6e4f5bb0A39eBaEb2335e3aF606057`. The details of the deployed smart contract are listed in Table 1.

FIELD	VALUE
Contract Name	LegendOfRPS
Contract Address	0x57f251706a6e4f5bb0A39eBaEb2335e3aF606057
Compiler Version	v0.8.7+commit.e28d00a7
Optimization Enabled	No with 200 runs
Explorer	*https://bscscan.com/address/0x57f251706a6e4f5bb0A39eBaEb2335e3aF606057*

Table 1. The deployed smart contract details

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 2. Severity levels

1.4. Disclaimer

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

The Legend of RPS Token was written in `Solidity` language, with the required version to be `^0.8.4`.

2.1.1. LegendOfRPS contract

`LegendOfRPS` is an ERC20 token contract that extends `ERC20Burnable`, `` `@ERC20 ``, `AccessControl` and `Pausable` contracts. With `AccessControl`, by default, the `ADMIN` is the contract deployer. `ERC20Burnable` allows token holders to destroy both their own tokens and those that they have an allowance for.

The `ADMIN` role can pause/unpause contract using `Pausable` contract. Also, The `ADMIN` role can `lockAddress` with an amount in a duration time. Locked users must wait until `locktimeEnd` to transfer their tokens to other addresses.

Also, the `ADMIN` role can use `mint` function to mint unlimited tokens.

2.2. Findings

During the audit process, the audit team found no vulnerability in the given version of the Legend of RPS Token.

2.2.1. Periodic - Token owner can mint unlimited amount of tokens INFORMATIVE

The token contract contains `mint` function for `ADMIN` role so the `ADMIN` role of the contract can `mint` unlimited amount of tokens without any cap.

APPENDIX

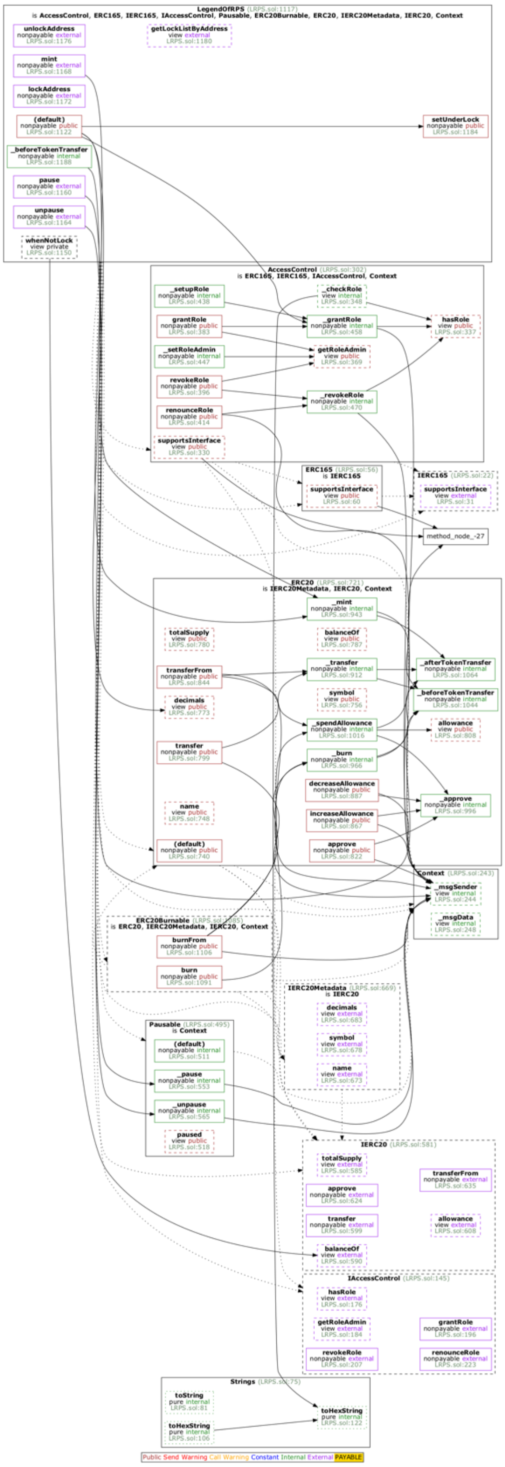
LegendOfRPS (LRPS.sol:1117)
is AccessControl, ERC165, IERC165, IAccessControl, Pausable, ERC20Burnable, ERC20, IERC20Metadata, IERC20, Context
unlockAddress nonpayable external LRPS.sol:1176
getLockListByAddress view external LRPS.sol:1180
mint nonpayable external LRPS.sol:1168
lockAddress nonpayable external LRPS.sol:1172
(default) nonpayable public LRPS.sol:1122
setUnderLock nonpayable public LRPS.sol:1184
_beforeTokenTransfer nonpayable internal LRPS.sol:1188
pause nonpayable external LRPS.sol:1160
unpause nonpayable external LRPS.sol:1164
whenNotLock view private LRPS.sol:1150
AccessControl (LRPS.sol:302)
is ERC165, IERC165, IAccessControl, Context
_setupRole nonpayable internal LRPS.sol:438
_checkRole view internal LRPS.sol:348
grantRole nonpayable public LRPS.sol:383
_grantRole nonpayable internal LRPS.sol:458
hasRole view public LRPS.sol:337
_setRoleAdmin nonpayable internal LRPS.sol:447
getRoleAdmin view public LRPS.sol:369
revokeRole nonpayable public LRPS.sol:396
_revokeRole nonpayable internal LRPS.sol:470
renounceRole nonpayable public LRPS.sol:414
supportsInterface view public LRPS.sol:330
ERC165 (LRPS.sol:56)
is IERC165
supportsInterface view public LRPS.sol:60
IERC165 (LRPS.sol:22)
supportsInterface view external LRPS.sol:31
method_node_-27
ERC20 (LRPS.sol:721)
is IERC20Metadata, IERC20, Context
_mint nonpayable internal LRPS.sol:943
totalSupply view public LRPS.sol:780
balanceOf view public LRPS.sol:787
transferFrom nonpayable public LRPS.sol:844
_transfer nonpayable internal LRPS.sol:912
_afterTokenTransfer nonpayable internal LRPS.sol:1064
decimals view public LRPS.sol:773
symbol view public LRPS.sol:756
_beforeTokenTransfer nonpayable internal LRPS.sol:1044
_spendAllowance nonpayable internal LRPS.sol:1016
allowance view public LRPS.sol:808
transfer nonpayable public LRPS.sol:799
_burn nonpayable internal LRPS.sol:966
decreaseAllowance nonpayable public LRPS.sol:887
_approve nonpayable internal LRPS.sol:996
name view public LRPS.sol:748
increaseAllowance nonpayable public LRPS.sol:867
(default) nonpayable public LRPS.sol:740
approve nonpayable public LRPS.sol:822
Context (LRPS.sol:243)
_msgSender view internal LRPS.sol:244
_msgData view internal LRPS.sol:248
ERC20Burnable (LRPS.sol:1085)
is ERC20, IERC20Metadata, IERC20, Context
burnFrom nonpayable public LRPS.sol:1106
burn nonpayable public LRPS.sol:1091
IERC20Metadata (LRPS.sol:669)
is IERC20
decimals view external LRPS.sol:683
symbol view external LRPS.sol:678
name view external LRPS.sol:673
Pausable (LRPS.sol:495)
is Context
(default) nonpayable internal LRPS.sol:511
_pause nonpayable internal LRPS.sol:553
_unpause nonpayable internal LRPS.sol:565
paused view public LRPS.sol:518
IERC20 (LRPS.sol:581)
totalSupply view external LRPS.sol:585
transferFrom nonpayable external LRPS.sol:635
approve nonpayable external LRPS.sol:624
transfer nonpayable external LRPS.sol:599
allowance view external LRPS.sol:608
balanceOf view external LRPS.sol:590
IAccessControl (LRPS.sol:145)
hasRole view external LRPS.sol:176
getRoleAdmin view external LRPS.sol:184
grantRole nonpayable external LRPS.sol:196
revokeRole nonpayable external LRPS.sol:207
renounceRole nonpayable external LRPS.sol:223
Strings (LRPS.sol:75)
toString pure internal LRPS.sol:81
toHexString pure internal LRPS.sol:122
toHexString pure internal LRPS.sol:106
Public Send Warning Call Warning Constant Internal External PAYABLE

Image 1. LRPS contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Sep 08, 2022*	Public Report	Verichains Lab

Table 3. Report versions history