verichains

SECURITY AUDIT OF

RUN TOGETHER TOKEN V2 SMART CONTRACT

Public Report

Jul 27, 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 Jul 27, 2022. We would like to thank the Run Together 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 Run Together Token v2 Smart Contract. 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 Run Together Token v2 Smart Contract

Run Together is built around the activities: walking and jogging - together every day, which is essential for everyone.

Users can equip themselves with NFT items which are divided in forms of:

- Sneakers
- Sportwears
- Glasses... and other accessories, etc.

Users can earn valuable tokens in the game as well as trade them for profit by walking, jogging, and running outside.

Run Together, which includes a Game-Fi component, aims to motivate millions of people to live healthier lifestyles, to fight the climate change, and connect people through Web 3.0 networks, while also utilizing the App's Social-Fi component to promote user-generated Web 3.0 content.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Run Together Token v2 Smart Contract.

The Run Together Token v2 Smart Contract was deployed on Binance Smart Chain Mainnet at address `0xc643E83587818202E0fFf5eD96D10Abbc8Bb48e7`. The details of the deployed smart contract are listed in table below.

FIELD	VALUE
Contract Name	RunTogetherToken
Contract Address	0xc643E83587818202E0fFf5eD96D10Abbc8Bb48e7
Compiler Version	v0.8.15+commit.e14f2714
Optimization Enabled	No with 200 runs
Explorer	*https://bscscan.com/address/0xc643E83587818202E0fFf5eD96D10Abbc8Bb48e7*

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 Run Together Token v2 Smart Contract was written in `Solidity` language, with the required version to be `^0.8.4`. The source code was written based on OpenZeppelin's library.

The Run Together Token v2 Smart Contract is an ERC20 token contract which extends `ERC20`, `ERC20Burnable`, `ERC20Snapshot`, `AccessControl` and `Pausable` contracts. `AccessControl` allows the contract to implement role-based access control mechanisms which add token owner (contract deployer) `DEFAULT_ADMIN_ROLE`, `PAUSER_ROLE` and `SNAPSHOT_ROLE` roles. Users who have `PAUSER_ROLE` can pause/unpause the contract using `Pausable` contract, users can only transfer tokens when the contract is not paused. The `PAUSER_ROLE` role can also blacklist addresses, blacklisted addresses can't send/receive tokens.

The contract pre-minted all 250 million tokens to the Contract Owner when deployed.

This table lists some properties of the ERC20 token contract (as of the report writing time).

PROPERTY	VALUE
Name	Run Together Token
Symbol	RUN
Decimals	18
Max Supply	250,000,000 ($x10^{18}$) Note: the number of decimals is 18, so the total representation token will be 250,000,000 or 250 million.

Table 3. The Run Together Token v2 Smart Contract properties

2.2. Findings

During the audit process, the audit team found no vulnerability in the given version of Run Together Token v2 Smart Contract.

APPENDIX

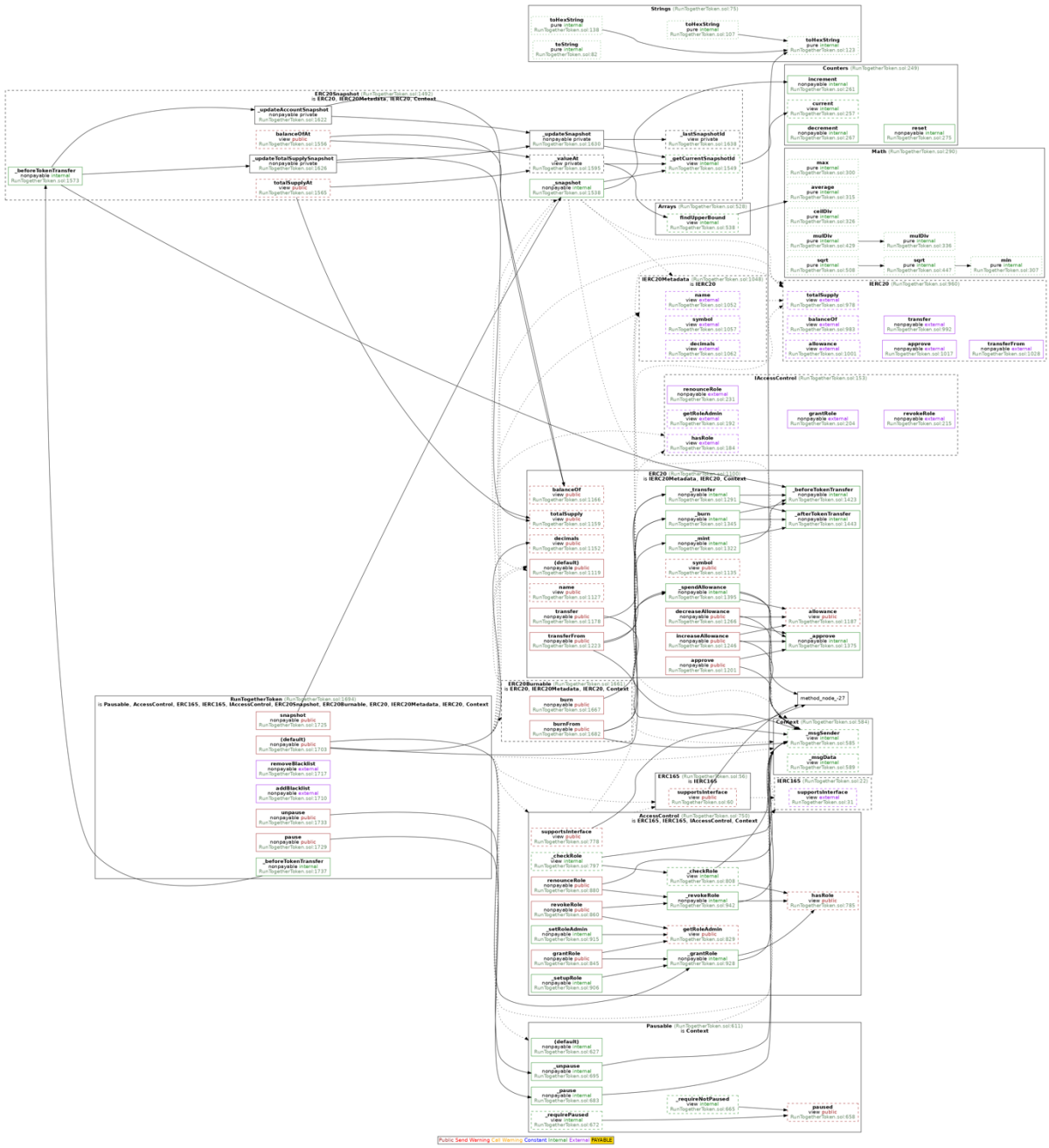

Image 1. The Run Together Token v2 Smart Contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Jul 27, 2022*	Public Report	Verichains Lab

Table 4. Report versions history