verichains

SECURITY AUDIT OF

YOGAIN TOKEN SMART CONTRACT

Public Report

May 02, 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 May 02, 2022. We would like to thank the Yogain 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 Yogain Token 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 Yogain Token Smart Contract

Yogain is a Web3 lifestyle app with Social-Fi and Game-Fi elements. Users are equipped with NFT Yoga Mat – practice yoga poses to earn $YOG, which can be used to level up and mint new mats. Players can choose to lease or sell their NFT mats on the in-app Marketplace; users' $YOG earnings are stored in the in-app Wallet, which has a built-in Swap function.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Yogain Token Smart Contract. Tt was conducted on commit 610a749d727e9e0a1ea19c5ebd5704256e1295dd from git repository *https://github.com/yogainteam/yogain-contract*.

The following files were made available in the course of the review:

SHA256 Sum	File
784671f90b9f1afa705d79f9da6e7478297d5fa536010435238bb7ef6ceb123a	**Yogain.sol**

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 1. 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 Yogain Token Smart Contract was written in Solidity language, with the required version to be ^0.8.0. The source code was written based on OpenZeppelin's library.

2.1.1. Yogain contract

This is ERC20 token contract in the Yogain Token Smart Contract, which extends ERC20Payable, ERC20Burnable and Withdrawable contracts. With Withdrawable, WITHDRAWER_ROLE role can withdraw all tokens, NFT (ERC20, ERC721, ERC1155) and native tokens from this contract.

ERC20Payable supports user paying invoices with this token. Invoices are generated and signed in backend server, users can only pay for a valid signature invoice.

The total supply of this token is 100,000,000 ($x10^{18}$) (Note: the number of decimals is 18, so the total representation token will be 100,000,000 or 100 million.)

2.2. Findings

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

APPENDIX

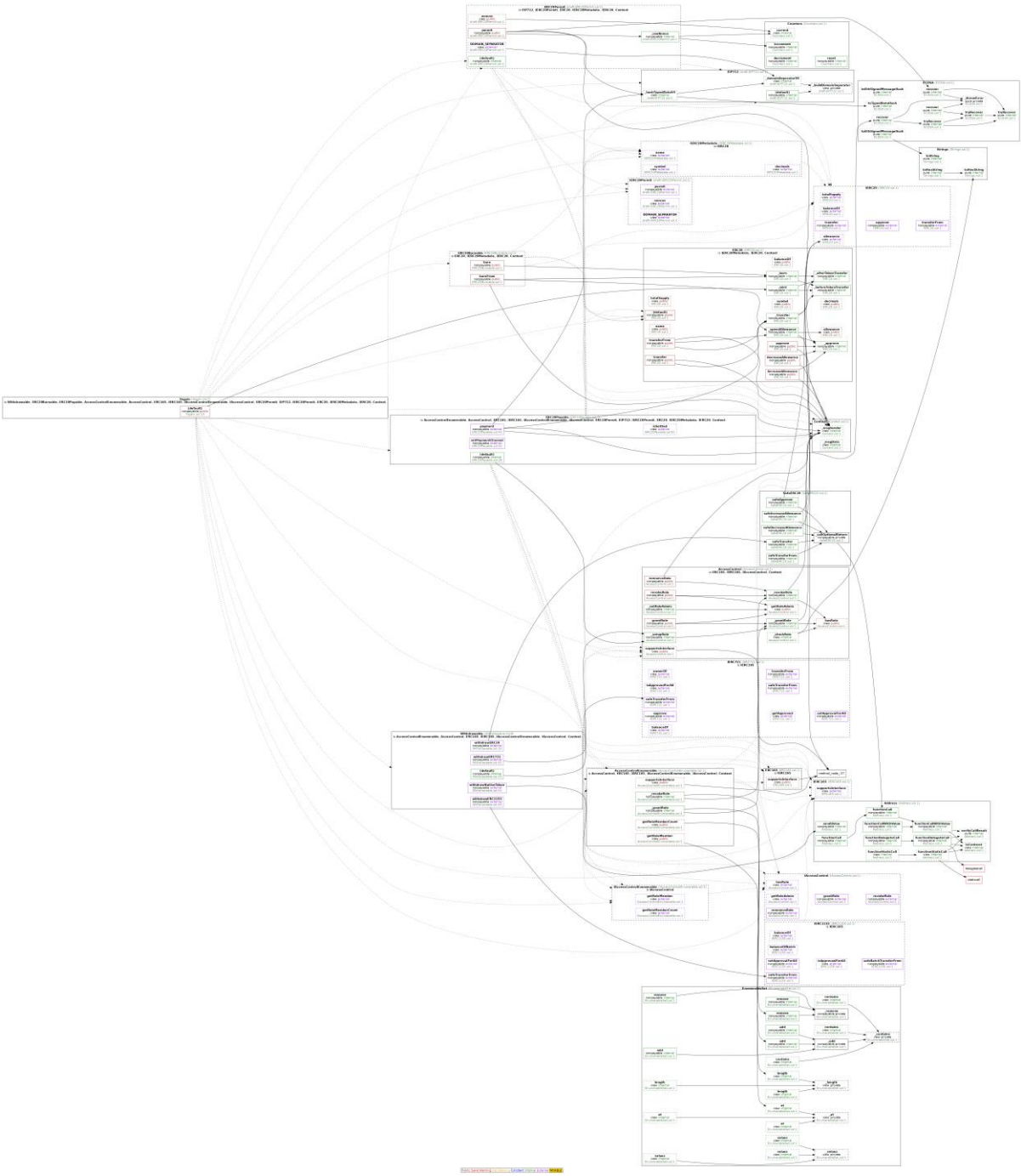

Image 1. Yogain Token Smart Contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*May 02, 2022*	Public Report	Verichains Lab

Table 2. Report versions history