verichains

SECURITY AUDIT OF

SDRCOIN SMART CONTRACT

Public Report

Jan 07, 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 Jan 07, 2022. We would like to thank the Sdrcoin 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 Sdrcoin 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 Sdrcoin Smart Contract

The underlying financial instrument of Special Drawing Rights Coin is the Special Drawing Rights (SDR). SDRs in its non-blockchain format refer to an international type of monetary reserve currency created by the International Monetary Fund (IMF) in 1969 that operates as a supplement to the existing money reserves of member countries. Created in response to concerns about the limitations of gold and dollars as the sole means of settling international accounts, SDRs augment international liquidity by supplementing the standard reserve currencies. By applying blockchain technology and the principles of decentralised finance (DeFi), SDR can now be used digitally as the SDR coin.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Sdrcoin Smart Contract.

The audited contract is the Sdrcoin Smart Contract that deployed on Ethereum Mainnet at address 0xbf0de2f2d2d7b361366333f4ff054c8e6ee2c46a. The details of the deployed smart contract are listed in Table 1.

FIELD	VALUE
Contract Name	StandardERC20
Contract Address	0xbf0de2f2d2d7b361366333f4ff054c8e6ee2c46a
Compiler Version	v0.8.7+commit.e28d00a7
Optimization Enabled	Yes with 200 runs
Explorer	*https://etherscan.io/address/0xbf0de2f2d2d7b361366333f4ff054c8e6ee2c46a*

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 Sdrcoin Smart Contract was written in Solidity language, with the required version to be 0.8.0.

The contract inherits ERC20Decimals contract which extends the Openzeppelin ERC20 contract for setting the decimals value in the constructor.

Table 2 lists some properties of the audited Sdrcoin Smart Contract (as of the report writing time).

PROPERTY	VALUE
Name	Special Drawing Rights
Symbol	SDR
Decimals	2
Total Supply	90,000,000,000 ($x10^2$) Note: the number of decimals is 2, so the total representation token will be 90,000,000,000 or 90 billion.

Table 3. The Sdrcoin Smart Contract properties

2.2. Findings

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

APPENDIX

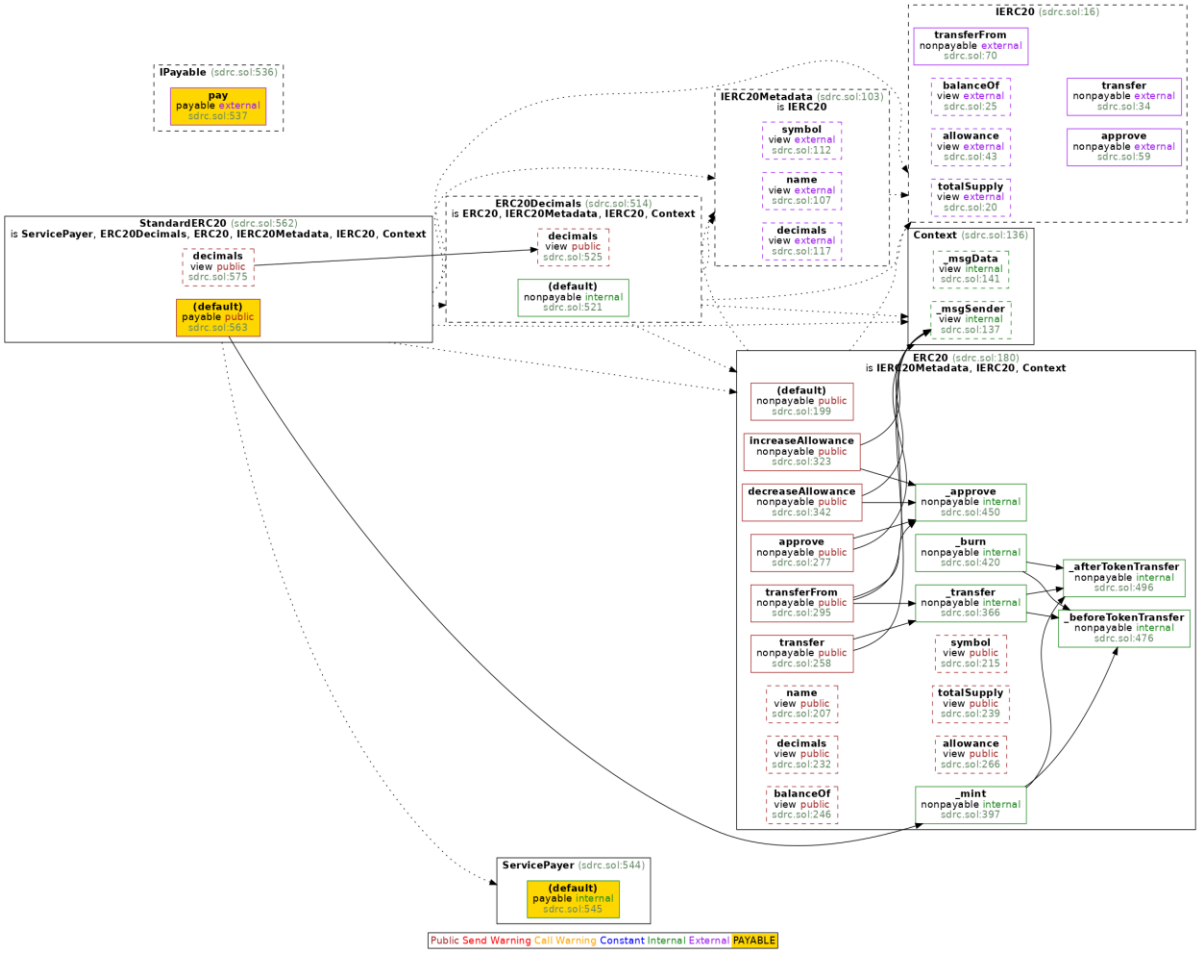

Image 1. Sdrcoin Smart Contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Jan 07, 2022*	Public Report	Verichains Lab

Table 4. Report versions history