SECURITY AUDIT OF

STE TOKEN SMART CONTRACT

Public Report

Jun 01, 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 Jun 01, 2022. We would like to thank the STE 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 STE 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 issue in the smart contract code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About STE Token Smart Contract

Stretch To Earn is a fitness app, both to improve health and to be able to earn up to 100,000 dollars on your body movements. STE was born to inspire millions of people to live a healthy lifestyle and motivate human beings by encouraging users to exercise daily and awarding them worthy rewards.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the STE Token Smart Contract. It was conducted on the source code provided by the STE team.

The audited contracts are the STE Token Smart Contract that deployed on Binance Smart Chain. The details of the deployed smart contract are listed in Table 1.

FIELD	VALUE
Contract Name	CoinToken
Contract Address	0xA4CcaE9F2bD6195053D675e9a09B1d894695Ca0F
Compiler Version	v0.8.5+commit.a4f2e591
Optimization Enabled	Yes with 200 runs
Explorer	*https://bscscan.com/token/0xA4CcaE9F2bD6195053D675e9a09B1d894695Ca0F*

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 STE Token Smart Contract was written in Solidity language, with the required version to be 0.8.0. Almost all source codes in the STE Token Smart Contract are imported from OpenZeppelin contracts.

CoinTokens contract extends ERC20, ERC20Burnable, TokenRecover, ERC20Decimals, ERC20Mintable, ERC1363 and Ownable abstract contract. With Ownable, by default, Token Owner is contract deployer but he can transfer ownership to another address at any time. ERC20Burnable allows token holders to destroy both their own tokens and those that they have an allowance for.

The ERC1363 is an implementation of the ERC-1363 Payable Token that defines a token interface for ERC-20 tokens that supports executing recipient code after transfer or transferFrom, or spender code after approve.

With ERC20Mintable, the owner of the contract may mint unlimited tokens. Therefore, the totalSupply value can be changed by this function.

The contract also extends TokenRecover abstract contract which allows the owner to withdraw any tokens in the contract.

Notice:

The owner of the contract was set to 0x000000000000000000000000000000000000dEaD address. Currently, no one controls this address, so all of the owner features are disabled (including mint).

References:

- *https://bscscan.com/tx/0x374df332c023f05fd3e4662a5da7ddba6ded5efb6eea68a3cb2646cec6185f94*

This table lists some properties of the deployed STE Token Smart Contract (as of the report writing time).

PROPERTY	VALUE
Name	Stretch To Earn
Symbol	STE
Decimals	18

PROPERTY	VALUE
Total Supply	10,000,000 (x10^{18}) Note: the number of decimals is 18, so the total representation token will be 10,000,000 or 10 million.

Table 3. The deployed STE Token Smart Contract properties

2.2. Findings

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

APPENDIX

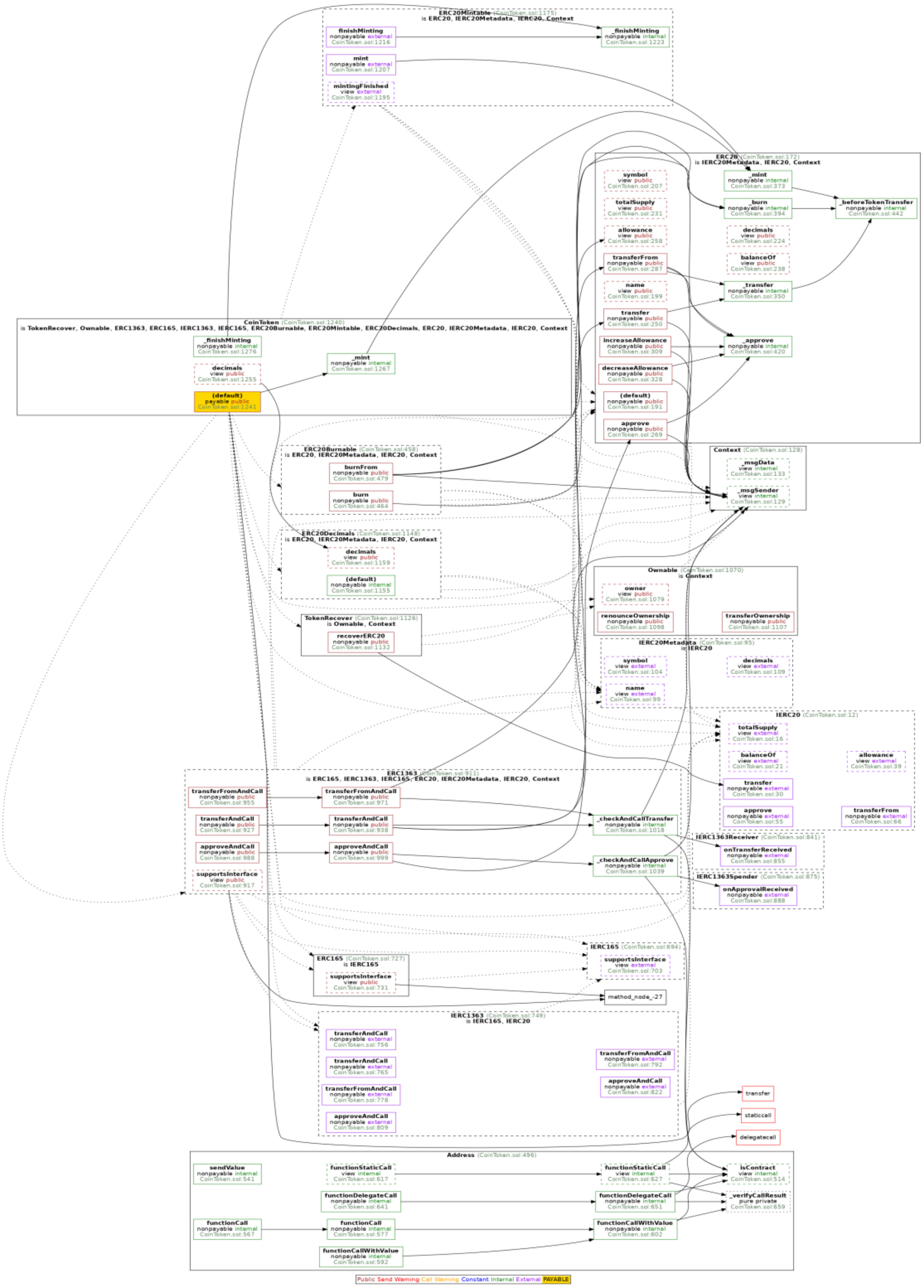
ERC20Mintable (CoinToken.sol:1175) is ERC20, IERC20Metadata, IERC20, Context
finishMinting nonpayable external CoinToken.sol:1216
mint nonpayable external CoinToken.sol:1207
mintingFinished view external CoinToken.sol:1195
_finishMinting nonpayable internal CoinToken.sol:1223
ERC20 (CoinToken.sol:172) is IERC20Metadata, IERC20, Context
symbol view public CoinToken.sol:207
totalSupply view public CoinToken.sol:231
allowance view public CoinToken.sol:258
transferFrom nonpayable public CoinToken.sol:287
name view public CoinToken.sol:199
transfer nonpayable public CoinToken.sol:250
increaseAllowance nonpayable public CoinToken.sol:309
decreaseAllowance nonpayable public CoinToken.sol:328
(default) nonpayable public CoinToken.sol:191
approve nonpayable public CoinToken.sol:269
_mint nonpayable internal CoinToken.sol:373
_burn nonpayable internal CoinToken.sol:394
decimals view public CoinToken.sol:224
balanceOf view public CoinToken.sol:238
_transfer nonpayable internal CoinToken.sol:350
_approve nonpayable internal CoinToken.sol:420
_beforeTokenTransfer nonpayable internal CoinToken.sol:442
CoinToken (CoinToken.sol:1240) is TokenRecover, Ownable, ERC1363, ERC165, IERC1363, IERC165, ERC20Burnable, ERC20Mintable, ERC20Decimals, ERC20, IERC20Metadata, IERC20, Context
_finishMinting nonpayable internal CoinToken.sol:1276
decimals view public CoinToken.sol:1255
(default) payable public CoinToken.sol:1241
_mint nonpayable internal CoinToken.sol:1267
Context (CoinToken.sol:128)
_msgData view internal CoinToken.sol:133
_msgSender view internal CoinToken.sol:129
ERC20Burnable (CoinToken.sol:458) is ERC20, IERC20Metadata, IERC20, Context
burnFrom nonpayable public CoinToken.sol:479
burn nonpayable public CoinToken.sol:464
ERC20Decimals (CoinToken.sol:1148) is ERC20, IERC20Metadata, IERC20, Context
decimals view public CoinToken.sol:1159
(default) nonpayable internal CoinToken.sol:1155
Ownable (CoinToken.sol:1070) is Context
owner view public CoinToken.sol:1079
renounceOwnership nonpayable public CoinToken.sol:1098
transferOwnership nonpayable public CoinToken.sol:1107
TokenRecover (CoinToken.sol:1126) is Ownable, Context
recoverERC20 nonpayable public CoinToken.sol:1132
IERC20Metadata (CoinToken.sol:95) is IERC20
symbol view external CoinToken.sol:104
decimals view external CoinToken.sol:109
name view external CoinToken.sol:99
IERC20 (CoinToken.sol:12)
totalSupply view external CoinToken.sol:16
balanceOf view external CoinToken.sol:21
allowance view external CoinToken.sol:39
transfer nonpayable external CoinToken.sol:30
approve nonpayable external CoinToken.sol:55
transferFrom nonpayable external CoinToken.sol:66
ERC1363 (CoinToken.sol:911) is ERC165, IERC1363, IERC165, ERC20, IERC20Metadata, IERC20, Context
transferFromAndCall nonpayable public CoinToken.sol:955
transferAndCall nonpayable public CoinToken.sol:927
approveAndCall nonpayable public CoinToken.sol:988
supportsInterface view public CoinToken.sol:917
transferFromAndCall nonpayable public CoinToken.sol:971
transferAndCall nonpayable public CoinToken.sol:938
approveAndCall nonpayable public CoinToken.sol:999
_checkAndCallTransfer nonpayable internal CoinToken.sol:1018
_checkAndCallApprove nonpayable internal CoinToken.sol:1039
IERC1363Receiver (CoinToken.sol:841)
onTransferReceived nonpayable external CoinToken.sol:855
IERC1363Spender (CoinToken.sol:875)
onApprovalReceived nonpayable external CoinToken.sol:888
IERC165 (CoinToken.sol:694)
supportsInterface view external CoinToken.sol:703
ERC165 (CoinToken.sol:727) is IERC165
supportsInterface view public CoinToken.sol:731
method_node_-27
IERC1363 (CoinToken.sol:749) is IERC165, IERC20
transferAndCall nonpayable external CoinToken.sol:756
transferAndCall nonpayable external CoinToken.sol:765
transferFromAndCall nonpayable external CoinToken.sol:778
approveAndCall nonpayable external CoinToken.sol:809
transferFromAndCall nonpayable external CoinToken.sol:792
approveAndCall nonpayable external CoinToken.sol:822
transfer
staticcall
delegatecall
Address (CoinToken.sol:496)
sendValue nonpayable internal CoinToken.sol:541
functionStaticCall view internal CoinToken.sol:617
functionDelegateCall nonpayable internal CoinToken.sol:641
functionCall nonpayable internal CoinToken.sol:567
functionCall nonpayable internal CoinToken.sol:577
functionCallWithValue nonpayable internal CoinToken.sol:592
functionStaticCall view internal CoinToken.sol:627
functionDelegateCall nonpayable internal CoinToken.sol:651
functionCallWithValue nonpayable internal CoinToken.sol:602
isContract view internal CoinToken.sol:514
_verifyCallResult pure private CoinToken.sol:659
Public Send Warning Call Warning Constant Internal External PAYABLE

Image 1. STE Token Smart Contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Mar 31, 2022*	Public Report	Verichains Lab
1.1	*Jun 01, 2022*	Public Report	Verichains Lab

Table 4. Report versions history