verichains

SECURITY AUDIT OF

CAPSHORT SMART CONTRACT

Public Report

Apr 27, 2023

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 was prepared by Verichains Lab on Apr 27, 2023. We would like to thank the Capshort 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 Capshort 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 Capshort Smart Contract

The DAO, which will be hosted in the Capshort ecosystem, is envisioned as a platform where users can collectively vote for new features and activities to be undertaken. Through this democratic process, the community will have a say in shaping the direction of the Capshort Center. The DAO is not just a voting platform but also a tool for empowering the community members to participate actively in the decision-making process of the ecosystem.

In the Capshort ecosystem, the Capshort token (CAP) plays a crucial role in governance. CAP is the native cryptocurrency of the ecosystem and is used to vote on proposals and decisions within the DAO. CAP holders can influence the direction of the platform by proposing and voting on changes, upgrades, and improvements. The more CAP a user holds, the greater their voting power within the DAO.

1.2. Audit scope

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

Repository Link	Audit commit
https://github.com/Capshort-World/smart-contract/tree/main/contracts	`6014da1b80ffd705767d5da0807f39e40d7af7cd`

Table 1. Audit Scope

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 were 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

2.1.1. DAO.sol

The purpose of this contract serve as a DAO (decentralized autonomous organization) that allows for proposal submission, voting, and execution in a decentralized manner. The inheritance from several other contracts suggests that this contract is designed to be customizable and modular, allowing developers to adjust the parameters of the DAO to fit their specific use cases.

2.1.2. Timelock.sol

The contract has several functions to check if an operation is pending, ready, or done, to get the timestamp at which an operation becomes ready, and to get whether an operation is registered. The contract can also receive ether.

2.1.3. Vesting.sol

The Vesting contract defines the rules and logic for vesting tokens for a specified beneficiary address. The contract allows for the configuration of multiple vesting schedules with different parameters such as the start time, cliff period, and vesting duration. These schedules can be added, edited, or deleted by the contract owner.

When a new vesting schedule is added for a beneficiary, the contract records the total amount of tokens to be vested and the specific time-based distribution of these tokens. Once the start time has passed, the beneficiary can begin claiming their vested tokens in installments based on the vesting schedule until the total amount has been distributed.

2.1.4. CAP.sol

The contract "CapToken" is an implementation of the ERC20BaseVote standard. It is designed to create a new token called "Capshort Token" (CAP), with an initial supply of 4 billion tokens. The constructor function takes an argument "tokenHolder" which is the address of the account that will receive the initial token supply.

Once the contract is deployed, the "_mint" function is called to create the initial token supply and transfer it to the token holder's address. The function takes two arguments: the first is the address of the recipient (in this case, the token holder), and the second is the number of tokens to be minted, which is calculated as 4 billion multiplied by 10^18 (the standard unit of measurement for Ethereum tokens).

2.2. Findings

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

2.3. Additional notes and recommendations

2.3.1. [INFORMATIVE] Compiling Error: Missing brace (}) in Code

Positions:

- L1221#Timelock.sol
- L4747#DAO.sol

RECOMMENDATION

To fix the code and make it work, you need to add a closing curly brace (}) in the line that was missing it.

UPDATES

- *Apr 27, 2023*: This issue has been acknowledged and fixed by the Capshort team.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Apr 21, 2023*	Public Report	Verichains Lab
1.1	*Apr 27, 2023*	Public Report	Verichains Lab

Table 3. Report versions history