SECURITY AUDIT OF

DEFIHORSE TOKEN VESTING SMART CONTRACT

Public Report

Mar 31, 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 Mar 31, 2022. We would like to thank the DeFiHorse 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 DeFiHorse Token Vesting 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 **5**

1.1. About DeFiHorse Token Vesting Smart Contract **5**

1.2. Audit scope **5**

1.3. Audit methodology **5**

1.4. Disclaimer **6**

2. AUDIT RESULT **7**

2.1. Overview **7**

2.2. Contract code **7**

2.2.1. DFHVestingToken contract 7

2.3. Findings **7**

2.4. Additional notes and recommendations **7**

2.4.1. Unnecessary usage of SafeMath library in Solidity 0.8.0+ INFORMATIVE 7

3. VERSION HISTORY **8**

1. MANAGEMENT SUMMARY

1.1. About DeFiHorse Token Vesting Smart Contract

The DeFiHorse is a metaverse e-sport game based on Blockchain technology and NFTs, empowering the players and creators to the next level of the horse race.

The game brings you the gorgeous legendary horses to enter the limitless cyberpunk horserace. Overwhelming yourself in every minute of the battle, using your personal and teamwork skills to earn token rewards, and you will end up with a 1,000,000 dollar prize pool.

The DeFiHorse Token Vesting is a contract which support the DeFiHorse team releasing the token was bought by the investors.

1.2. Audit scope

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

It was conducted on commit fec11987880f2e857dcc155a1dda7e34b4dbca81 from git repository *https://github.com/DefiHorse/autovesting*.

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

SHA256 Sum	File
667f7ab913e357a12d24d4a89e6e2ba14a08655a0316c2c458a912f568fd23d1	**DFHVesting-v2.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

2.2. Contract code

The DeFiHorse Token Vesting 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.

The provided source codes consist of four contracts which inherit some contracts from OpenZeppelin.

2.2.1. DFHVestingToken contract

The DeFiHorse team uses this contract to release the tokens that investors bought in the past. The tokens of investors will be released in 2 phases. In the first phase - after the first cliff duration, the investors receive the number of tokens which corresponds to a number percent of bought tokens. In the second phase, tokens will be released following the linear logic every secondsPerSlice seconds.

The contract implements withdrawToken and withdrawTokenAll public functions which allow the owner of the contract may transfer any tokens that were included in the contract to any address wallet.

2.3. Findings

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

2.4. Additional notes and recommendations

2.4.1. Unnecessary usage of SafeMath library in Solidity 0.8.0+ INFORMATIVE

All safe math usages in the contract are for overflow checking, solidity 0.8.0+ already do that by default, the only usage of SafeMath now is to have a custom revert message which isn't the case in the auditing contracts. We suggest using normal operators for readability and gas saving.

RECOMMENDATION

We suggest changing all methods from SafeMath library to normal arithmetic operator.

3. VERSION HISTORY

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

Table 2. Report versions history