verichains

*SECURITY AUDIT OF*

# DEFIHORSE TOKEN SMART CONTRACT


**Public Report**

*Dec 27, 2021*


## 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 Dec 27, 2021. 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 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 DeFiHorse Token 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.

## 1.2. Audit scope

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

It was conducted on commit 3dd8b97d5029999cf4ac30bdcf719ef687760c69 from git repository *https://github.com/DefiHorse/DFHToken-PSC/*.

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

| SHA256 SUM | FILE |
|---|---|
| **84a9288fb21a9b016a2fd0accab522112382a589ed8435a7608d695abce38268** | DefiHorse.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

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

| PROPERTY | VALUE |
|---|---|
| **Name** | DefiHorse |
| **Symbol** | DFH |
| **Decimals** | 18 |
| **Total Supply** | 668,000,000 ($x10^{18}$)<br>Note: the number of decimals is 18, so the total representation token will be 668,000,000 or 668 million. |

*Table 2. The DeFiHorse Token Smart Contract properties*

The DeFiHorse Token Smart Contract was written in Solidity language, with the required version to be ^0.8.2.

HorseToken contract extends ERC20, ERC20Burnable, ERC20Snapshot, Ownable and Pausable contracts. 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. ERC20Snapshot help Token Owner take a snapshot of the balances and total supply at a time for later access.

Token Owner can pause/unpause contract using Pausable contract, user can only transfer unlocked tokens and only when contract is not paused.

The contract also takes tax when transfer tokens (default is 1%, owner can change it anytime).

## 2.2. Findings

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

## 2.3. Additional notes and recommendations

DeFiHorse fixed the code, according to Verichain's private report, in commit 332f62b8178fb2aded08932980c0870d831d0258.

### 2.3.1. Redundant require INFORMATIVE

In _transfer function, require(amount == sendAmount + taxAmount,"transfer: Tax val invalid"); is redundant because solidity 0.8.0+ already check for overflow by default.

```
function _transfer(
    address sender,
    address recipient,
    uint256 amount
)
internal
virtual
override(ERC20)
whenNotPaused
{

    {
        // default tax is 1% of every transfer
        uint256 taxAmount = (amount*_taxFee)/100;

        // default 99% of transfer sent to recipient
        uint256 sendAmount = amount-(taxAmount);
        require(
            amount == sendAmount + taxAmount,
            "transfer: Tax val invalid"
        );

        super._transfer(sender, _taxAddress, taxAmount);
        super._transfer(sender, recipient, sendAmount);
        amount = sendAmount;
    }
}
```

**RECOMMENDATION**

Consider removing require statement.

**UPDATES**

- *Dec 27, 2021*: This issue has been acknowledged and fixed by the DeFiHorse team.

### 2.3.2. Modifier onlyOwner is using inconsistently INFORMATIVE

Modifier onlyOwner is using inconsistently (onlyOwner and onlyOwner()) in the contract.

```
function pause() public onlyOwner {
    _pause();
}

function unpause() public onlyOwner {
    _unpause();
}

function mint(address to, uint256 amount) public onlyOwner {
    _mint(to, amount);
}

function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

function setTaxAddress(address taxAddress) external onlyOwner() {
    _taxAddress = taxAddress;
}
```

**RECOMMENDATION**

Using onlyOwner for all functions.

**UPDATES**

- *Dec 27, 2021*: This issue has been acknowledged and fixed by the DeFiHorse team.

### 2.3.3. _taxAddress visibility is internal INFORMATIVE

_taxAddress visibility is internal by default, so we can't call the contract to check for its value is correct or not.

```
address _taxAddress = 0x93A364BcA26F754DaBF95fC5Ac3cb62196b65590;
```

**RECOMMENDATION**

Consider making it public or add a getter to return its value.

```
address public _taxAddress = 0x93A364BcA26F754DaBF95fC5Ac3cb62196b65590;
```

**UPDATES**

- *Dec 27, 2021*: This issue has been acknowledged and fixed by the DeFiHorse team.

# Appendix

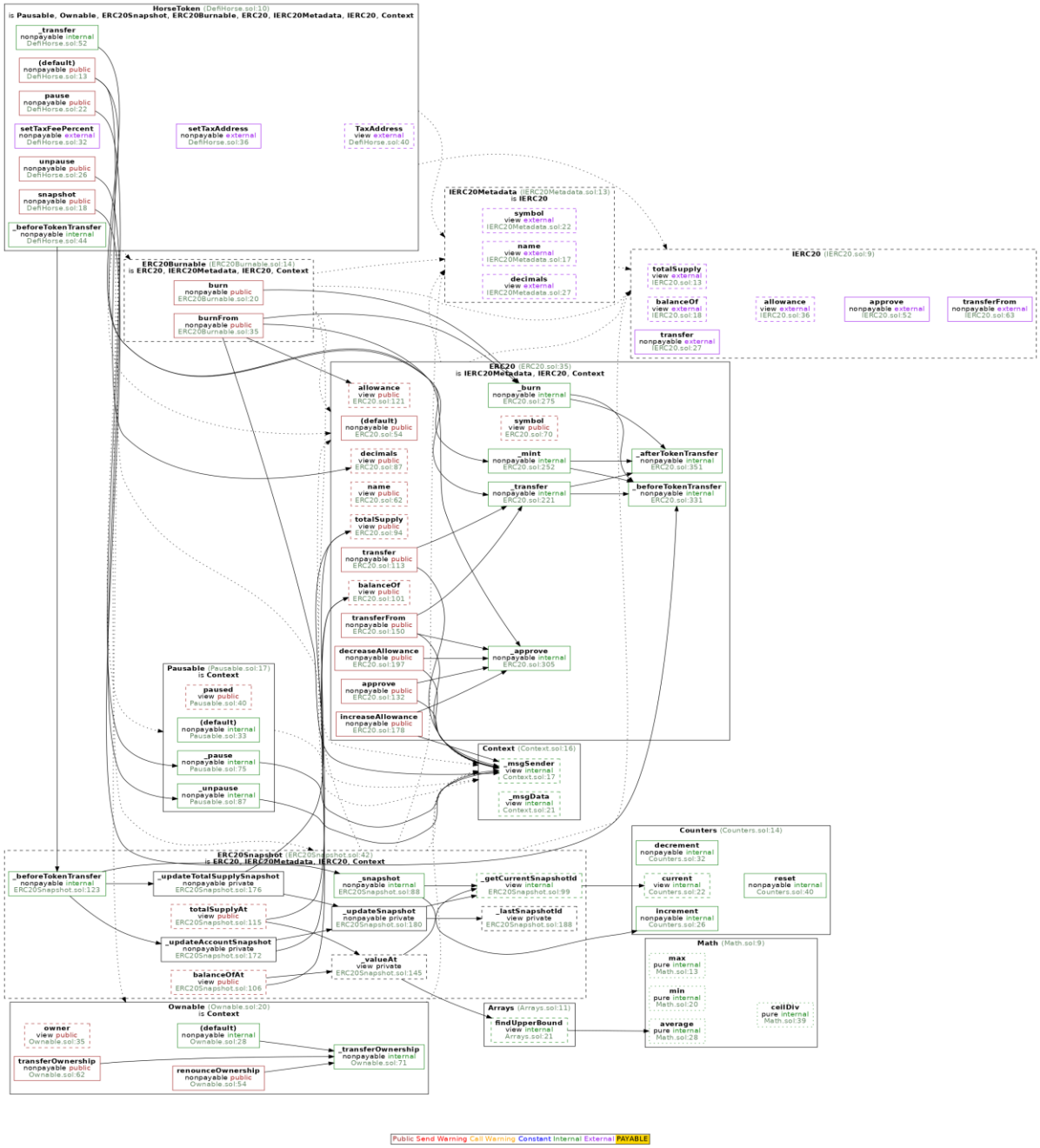


*Image 1. DeFiHorse Token Smart Contract call graph*

# 3. VERSION HISTORY

| Version | Date | Status/Change | Created by |
|---|---|---|---|
| **1.0** | *Dec 21, 2021* | Private Report | Verichains Lab |
| **1.1** | *Dec 27, 2021* | Public Report | Verichains Lab |

*Table 3. Report versions history*