



Advanced Manual

# Smart Contract Audit

March 20, 2022

Audit Requested by



**BLOCKCHAIN  
SECURITY & SMART  
CONTRACT AUDIT**

 <https://codeproof.xyz>

 [https://twitter.com/CodeProof\\_Labs](https://twitter.com/CodeProof_Labs)

 <https://t.me/CodeProof>

 [infor@codeproof.xyz](mailto:infor@codeproof.xyz)

# Table of Contents

## 1. Audit Summary

- 1.1 Audit Scope
- 1.2 Tokenomics
- 1.3 Source Code

## 2. Global Overview

- 3.1 Information Issues
- 3.2 Low-Risk Issues
- 3.3 Medium-Risk Issues
- 3.4 High-Risk Issues

## 3. Vulnerabilities Findings

## 4. Contract Privileges

- 5.1 Maximum Fee Limit Check
- 5.2 Contract Pausability Check
- 5.3 Max Transaction Amount Check
- 5.4 Exclude From Fees Check
- 5.5 Ability to Mint Check
- 5.6 Ability to Blacklist Check
- 5.7 Owner Privileges Check

## 5. Website Review

## 6. Certificate of Proof

## 7. Disclaimer

## Audit Summary

|                         |   |
|-------------------------|---|
| Project Name            | DERIFY  |
| Website                 | <a href="https://www.derify.finance/">https://www.derify.finance/</a> |
| Blockchain              | BNB Chain   |
| Smart Contract Language | Solidity  |
| Contract Address        | 0x89c1af791d7b4cf046dca8fa10a41dd2298a6a3f                            |
| Audit Method            | Static Analysis, Manual Review  |
| Date of Audit           | March 20, 2022  |
| Score of Audit          | 8 out of 10   |

This audit report was prepared by CodeProof's experts at the client's request. In this review, the results of static analysis and manual code review will be presented. The purpose of the audit is to see if functionality works as expected and identify potential security issues in smart contracts.

The information in this report should be used to understand the risks associated with smart contracts. This report can serve as a guide for the development team on how they can improve the contract by correcting identified issues.

## Audit Scope

CodeProof was commissioned to perform an audit based on the following code: <https://bscscan.com/token/0x4fdda510590117860b9d3db46efb8df002a831ce>

Note that when auditing, we only audit the code available at that URL. If the URL is not from any block explorer (mainnet), it may be subject to change. Always check the contract address on this audit report and compare it to the coins you are researching.

## Audit method

CodeProof's Manual Smart Contract Audit is an extensive systematic inspection and analysis of the smart contract code used to interact with the blockchain. This process is performed to find bugs, issues, and security holes in the code in order to suggest improvements and ways to fix them.

## Automatic Vulnerability Check

CodeProof uses software to check for common vulnerability issues in smart contracts. We use automated tools to scan contracts for security vulnerabilities such as integer overflows, integer underflows, out-of-gas conditions, unchecked transfers, etc.

## Manual Code Review

Co-insulting manual code review involves a human going through the source code line by line to find vulnerabilities. Manual code reviews help clarify the context of coding decisions. Automated tools are faster, but they cannot understand developer intent and general business logic consideration.

## Used Tools

Slither: Solidity static analysis framework

Remix: IDE Developer Tool





CWE: Common Weakness Enumeration

SWC: Smart Contract Weakness Classification and Test Cases

DEX: Testnet Blockchains

## Risk Classification

CodeProof uses specific vulnerability levels that indicate how serious a problem is. The higher the risk, the more rigorously it is recommended to correct errors before using the contract.

| Vulnerability Level   | Description   |
|---|---|
|  Information | Not in any way impair the functionality of the contract |
|  Low-Risk    | Won't cause any problems, but can be improved           |
|  Medium-Risk | Likely cause problems and recommended to adjust         |
|  High-Risk   | Definitely cause problems, needs to be adjusted         |

CodeProof has four states for each risk level. Below we briefly explain.

| Risk Status  | Description   |
|--------------|---|
| Total        | Total amount of issues within this category         |
| Pending      | Risks that have yet to be addressed by the team     |
| Acknowledged | The team is aware of the risks but not resolve them |
| Resolved     | The team has resolved and remedied the risk         |

## SWC Attack Analysis





The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It aligns loosely with the terminology and structure used in the Common Weakness Enumeration (CWE), while covering a wide range of vulnerability variants specific to smart contracts.

| Risk Status | Description                             | Status |
|-------------|---|--------|
| SWC-101     | Function Default Visibility             | Passed |
| SWC-101     | Integer Overflow and Underflow          | Passed |
| SWC-102     | Outdated Compiler Version               | Passed |
| SWC-103     | Floating Pragma                         | Passed |
| SWC-104     | Unchecked Call Return Value             | Passed |
| SWC-105     | Unprotected Ether Withdrawal Passed     | Passed |
| SWC-106     | Unprotected SELFDESTRUCT Instruction    | Passed |
| SWC-107     | Reentrancy                              | Passed |
| SWC-108     | State Variable Default Visibility       | Failed |
| SWC-109     | Uninitialized Storage Pointer Passed    | Passed |
| SWC-110     | Assert Violation                        | Passed |
| SWC-111     | Use of Deprecated Solidity Functions    | Passed |
| SWC-112     | Delegatecall to Untrusted Callee Passed | Passed |
| SWC-113     | DoS with Failed Call                    | Passed |
| SWC-114     | Transaction Order Dependence            | Passed |
| SWC-115     | Authorization through tx.origin Failed  | Failed |

# Global Overview







## Manual Code Review

In this audit report we will highlight the following issues:

| Vulnerability Level   | Total | Pending | Acknowledged | Resolved |
|---|-------|---------|--------------|----------|
|  Information | 0     | 0       | 0            | 0        |
|  Low-Risk    | 3     | 3       | 0            | 0        |
|  Medium-Risk | 1     | 1       | 0            | 0        |
|  High-Risk   | 0     | 0       | 0            | 0        |

## Centralization Risks

CodeProof checked the following privileges:




| Risk Status                  | Description  |
|------------------------------|--|
| Owner can mint?              |  Owner cannot mint new tokens                   |
| Owner can blacklist?         |  Owner cannot blacklist addresses               |
| Owner can set fees > 25%?    |  Owner cannot set the sell fee to 25% or higher |
| Owner can exclude from fees? |  Owner can exclude from fees                    |
| Owner can pause trading?     |  Owner cannot pause the contract                |
| Owner can set Max TX amount? |  Owner can set Max TX amount                    |

More owner privileges are listed later in the report.

## Maximum Fee Limit Check

| Error Code | Description                               |
|------------|---|
| CEN-01     | Centralization: Operator Fee Manipulation |

CodeProof tests if the owner of the smart contract can set the transfer, buy or sell fee to 25% or more. It is bad practice to set the fees to 25% or more, because owners can prevent healthy trading or even stop trading when the fees are set too high.


| Privilege Check | Description   |
|-----------------|---|
| Transfer Fee    |  Owner cannot set the transfer fee to 25% or higher |
| Buy Fee         |  Owner cannot set the buy fee to 25% or higher     |
| Sell Fee        |  Owner cannot set the sell fee to 25% or higher    |



## Contract Pausability Check

| Error Code | Description                          |
|------------|--------------------------------------|
| CEN-02     | Centralization: Operator Pausability |


CodeProof tests whether the owner of the smart contract has the ability to pause the contract. If this is the case, users can no longer interact with the smart contract; users can no longer trade the token.

| Privilege Check               | Description   |
|-------------------------------|---|
| Can owner pause the contract? |  Owner cannot pause the contract |

# Max Transaction Amount Check

| Error Code | Description                                       |
|------------|---|
| CEN-03     | Centralization: Operator Transaction Manipulation |

CodeProof tests whether the owner of a smart contract can set a maximum amount for a transaction. If a transaction exceeds this limit, the transaction will revert. Owners can block normal transactions if they abuse this feature.

| Privilege Check              | Description   |
|------------------------------|---|
| Owner can set Max TX amount? |  Owner can set Max TX amount |

## Function

CodeProof checked the following privileges:

```
function updateMaxTx(uint256 maxTXPercentage_base1000) external external  
onlyOwner  
require (maxTXPercentage_base1000 >= 5,"Cannot set max transaction less  
than 0.5%"); _maxTxAmount _(totalSupply * maxTXPercentage_base1000 ) //  
1000;
```

## Exclude From Fees Check

| Error Code | Description                        |
|------------|------------------------------------|
| CEN-04     | Centralization: Operator Exclusion |

CodeProof tests whether the owner of a smart contract can set a maximum amount for a transaction. If a transaction exceeds this limit, the transaction will revert. Owners can block normal transactions if they abuse this feature.

| Privilege Check              | Description  |
|------------------------------|--|
| Owner can set Max TX amount? | <input checked="" type="radio"/> Owner can set Max TX amount |

### Function

CodeProof checked the following privileges:

```
function updateMaxTx(uint256 maxTXPercentage_base1000) external external
onlyOwner
require (maxTXPercentage_base1000 >= 5,"Cannot set max transaction less
than 0.5%"); _maxTxAmount _(totalSupply * maxTXPercentage_base1000 ) //
1000;
```


## Ability To Mint Check

| Error Code | Description                              |
|------------|--|
| CEN-05     | Centralization: Operator Increase Supply |

CodeProof tests whether the owner of the smart contract can mint new tokens. If the contract contains a mint function, we refer to the total supply of tokens as non-fixed, allowing token owners to "mint" more whenever they want.

The mint function in the smart contract allows minting tokens at a later stage. A method to disable minting can also be added to irreversibly stop the minting process.

Minting tokens is done by sending a transaction that creates new tokens in the token smart contract. With the help of smart contract functionality, an unlimited number of tokens can be created without extra effort or money required.


| Privilege Check | Description  |
|-----------------|--|
| Can owner mint? |  Owner cannot mint new tokens |

## Ability To Blacklist Check

| Error Code | Description                                |
|------------|--|
| CEN-06     | Centralization: Operator Dissalows Wallets |

CodeProof tests whether the owner of a smart contract can blacklist accounts that interact with the smart contract. The blacklist method allows the contract owner to enter wallet addresses that are not allowed to interact with the smart contract.

This method can be abused by token owners to prevent some/all holders from trading tokens. However, blacklists can be fine for tokens that want to exclude certain addresses from interacting with smart contracts.

| Privilege Check | Description  |
|-----------------|--|
| Can owner mint? |  Owner cannot mint new tokens |

## Other Owner Privileges Check

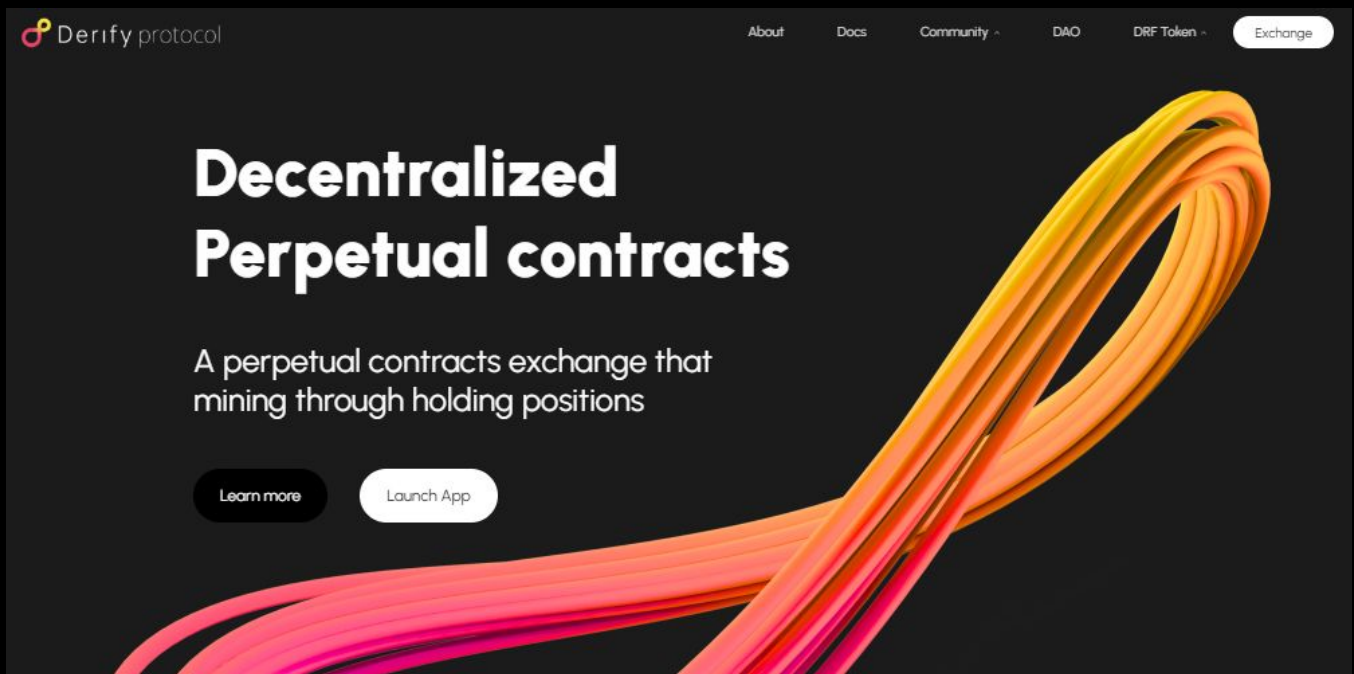
| Error Code | Description                         |
|------------|-------------------------------------|
| CEN-100    | Centralization: Operator Priviliges |

CodeProof lists all important contract methods which the owner can interact with.

- ⚠ Owner can open trading whenever he wants
- ⚠ Owner can withdraw tokens from the contract balance
- ⚠ Owner can set max holding balance
- ⚠ Owner can exclude addresses from max holding limit
- ⚠ Owner can exclude addresses from max transaction amount
- ⚠ Owner can set target liquidity level
- ⚠ Authorized wallets can trade before trading is opened

## Website Review

CodeProof checks websites completely manually, looking for visual, technical and textual errors. We also consider the security, speed and accessibility of the website. In short, a complete check to see if the site meets current web development industry standards.



| Type of Check           | Description                                  |
|-------------------------|--|
| Mobile friendly?        | ● The website is mobile friendly             |
| Contains jQuery errors? | ● The website does not contain jQuery errors |
| Is SSL secured?         | ● The website is SSL secured                 |



### **Audited by CodeProof**

Just like any other software, smart contracts come with security vulnerabilities. Therefore, a smart contract audit is necessary for ensuring that smart contracts are free of any security issues. Also, the audit will show where the smart contract can be optimized to ensure ideal levels of performance.



## Disclaimer

This audit report was prepared by CodeProof's experts at the client's request. In this review, the results of static analysis and manual code review will be presented. The purpose of the audit is to see if the functionality is working as expected and to identify potential security issues in the smart contract.

The information in this report should be used to understand the risks associated with smart contracts. This report can serve as a guide for the development team on how they can improve the contract by correcting identified issues.

CodeProof is not responsible if a project turns out to be a scam, rug pull or honeypot. We only provide detailed analysis for your own research. CodeProof is not responsible for any financial loss. There is no financial advice in this contract audit, please do your own research.

The information presented in this audit is for informational purposes only and should not be considered investment advice.

CodeProof does not endorse, recommend, support or advise investing in any project. CodeProof cannot be held responsible when a project turns out to be a rug pull, honeypot or scam.



End of report

# Smart Contract Audit

## BLOCKCHAIN SECURITY & SMART CONTRACT AUDIT

CodeProof is a pioneer in blockchain security, utilizing best-in-class manual security checks and AI technology to secure, develop and monitor blockchains, smart contracts, and Web3 apps

[CONTACT US](#)

 <https://codeproof.xyz>

 [https://twitter.com/CodeProof\\_Labs](https://twitter.com/CodeProof_Labs)

 <https://t.me/CodeProof>

 [infor@codeproof.xyz](mailto:infor@codeproof.xyz)