CodeProof

Advanced Manual

Smart Contract Audit

Sept. 7, 2023

Audit Requested by

Makalink

🌐 https://makalink.com/

CodeProof

Table of Contents

CodeProof

Audit Summary

Project Name	Makalink
Website	https://makalink.com/
Blockchain	Arbitrum
Smart Contract Language	Solidity
Contract Address	0xcc1E0F8481782F6f229BE7690D62d2aEcbf2a657
Audit Method	Static Analysis, Manual Review
Date of Audit	Sept. 7, 2023
Score of Audit	9 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

CodeProof

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-10l	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	Passed
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	Passed

CodeProof

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	5	5	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 infinitely?	🟢 Owner can mint, finish minting and recover
Owner can blacklist?	🟢 Owner cannot blacklist addresses
Enable airdrop address check?	🟢 No airdrop address check
Owner can pause trading?	🟢 Owner cannot pause the contract
Enable airdrop access control?	🟢 No airdrop access control
Anti-front-running measures?	🟡 No anti-front-running measures.

More owner privileges are listed later in the report.

Owner Centralization Risk

Error Code	Description
CEN-01	Centralization: Operator Increase Supply

The contract allows the owner to mint, finish minting, and recover tokens. This centralizes power in the hands of the contract owner, which could be a risk if the owner's account is compromised or if the owner acts maliciously. To mitigate this risk, it is recommended to implement a decentralized governance mechanism that requires consensus among multiple parties before executing critical functions.

Privilege Check	Description
Owner can mint infinitely?	🟢 Owner can mint, finish minting and recover

CodeProof

Ability To Blacklist Check

Error Code	Description
CEN-02	Centralization: Operator Disallow 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
Owner can blacklist?	🟢 Owner cannot blacklist addresses

Airdrop Address Check

Error Code	Description
CEN-03	No airdrop address check

In the Airdrop function, there are no validity checks for the input addresses. This may lead to tokens being sent to invalid addresses, resulting in a permanent loss of tokens. To address this issue, add a check in the Airdrop function to ensure the input addresses are not zero addresses.

Privilege Check	Description
Enable airdrop address check?	🟢 No airdrop address check

CodeProof

Contract Pausability Check

Error Code	Description
CEN-04	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

Airdrop Access Control

Error Code	Description
CEN-05	No airdrop access control

In In the current contract, anyone can call the Airdrop function. This may allow malicious users to slow down the contract's execution by making a large number of small transfers, which would affect the user experience for other users. To address this issue, consider making the Airdrop function callable only by the contract owner, or set a specific whitelist of addresses that are allowed to call the Airdrop function.

Privilege Check	Description
Enable airdrop access control?	🟢 No airdrop access control

Anti-front-running Measures Check

Error Code	Description
CEN-06	Front-running Attack Risk

CodeProof tests the contract does not implement any measures to prevent front-running attacks, where malicious actors observe pending transactions and attempt to profit from them by submitting their own transactions with higher gas prices. This could lead to a loss of funds for honest users and could be exploited by bots. To address this issue, it is recommended to implement measures such as a time delay or commit-reveal scheme to protect against front-running attacks.

Privilege Check	Description
Anti-front-running measures?	🟢 No anti-front-running measures.

CodeProof

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 mint new tokens whenever he wants

⚠ Owner can finish the minting process whenever he wants

⚠ Owner can can recover any ERC20 tokens accidentally sent to the contract

⚠ Owner can transfer the ownership to a new address.

CodeProof

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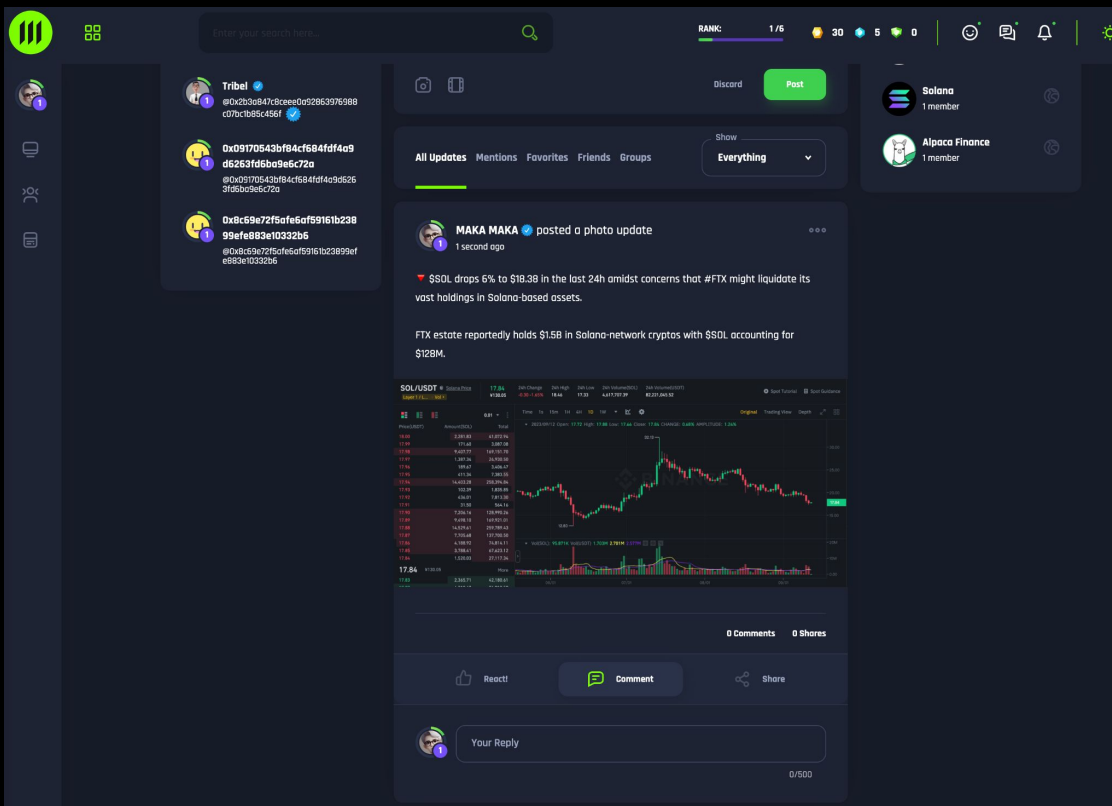
Type of Check	Description
Mobile friendly?	🟢 The website is mobile friendly
Contains jQuery errors?	🟢 The website does not contain jQuery errors
Is SSL secured?	🟢 The website is SSL secured

Makalink

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.

CodeProof

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.

CodeProof

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://t.me/CodeProof

✉️ infor@codeproof.xyz