Coinsult

Request your audit at **coinsult.net**

Advanced Manual

Smart Contract Audit

October 9, 2025

𝕏 x.com/coinsultaudits

t.me/coinsult_tg

Audit requested by

Linxir

0xd5f86184C99041CBefa63Dc55a11D67126581584

Global Overview

Manual Code Review

In this audit report we will highlight the following issues:

Vulnerability Level	Total	Pending	Acknowledged	Resolved
🔵 Informational	0	0	0	0
🟢 Low-Risk	0	0	0	0
🟠 Medium-Risk	1	0	0	1
🔴 Critical-Risk	1	0	0	1

Risk Classification

Coinsult uses certain vulnerability levels, these indicate how bad a certain issue is. The higher the risk, the more strictly it is recommended to correct the error before using the contract.

Vulnerability Level	Description
🔵 Informational	Does not compromise the functionality of the contract in any way
🟢 Low-Risk	Won't cause any problems, but can be adjusted for improvement
🟠 Medium-Risk	Will likely cause problems and it is recommended to adjust
🔴 Critical-Risk	Will definitely cause problems, this needs to be adjusted

Audit Summary

Project	
Website	https://www.linxir.com
Blockchain	Ethereum
Source Code	https://etherscan.io/address/0xd5f86184C99041CBefa63Dc55a11D67126581584#code
Contract Address	0xd5f86184C99041CBefa63Dc55a11D67126581584

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identified.

Audit Scope

Coinsult was commissioned to perform an audit based on the provided code.

Note that we only audited the code available to us on this URL at the time of the audit. If the URL is not from any block explorer (main net), it may be subject to change. Always check the contract address on this audit report and compare it to the token you are doing research for.

Audit Method

Coinsult's manual smart contract audit is an extensive methodical examination and analysis of the smart contract's code that is used to interact with the blockchain. This process is conducted to discover errors, issues and security vulnerabilities in the code in order to suggest improvements and ways to fix them.

Automated Vulnerability Check

Coinsult uses software that checks for common vulnerability issues within smart contracts. We use automated tools that scan the contract for security vulnerabilities such as integer-overflow, integer-underflow, out-of-gas-situations, unchecked transfers, etc.

Manual Code Review

Coinsult's manual code review involves a human looking at source code, line by line, to find vulnerabilities. Manual code review helps to clarify the context of coding decisions. Automated tools are faster but they cannot take the developer's intentions and general business logic into 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

Table of Contents

Notes:

Note Code	Description
CN-1	This audit is strictly limited to the Presale contract; please note that the associated Token contract was not reviewed and remains outside the scope of this engagement despite its heavy integration.

● Price depends on chainlink oracle (Resolved ✅)

Medium-Risk *- Will likely cause problems and it is recommended to adjust*

Error Code	Description
CER-M-1	If the Chainlink oracle stops updating, it will return an old "stale" price. Users could buy tokens much cheaper than intended, or the contract could overcharge them.

Code Snippet

```
function getLatestETHPrice() public view returns (uint256) {
        (, int256 price, , , ) = ethUsdPriceFeed.latestRoundData();
        require(price > 0, "Invalid price");
        return uint256(price) * 1e10;
    }
```

Recommendation

Validate the updatedAt timestamp (which can be retrieved from the latestRoundData function and ensure the data isn't too old.

● Impossible USDT Transfer (Resolved ✅)

Critical-Risk *- Will definitely cause problems, this needs to be adjusted*

Error Code	Description
CER-H-1	On Ethereum Mainnet, the USDT contract **does not return a boolean**. When USDT is called this way, the transaction will **always revert**, making it impossible for users to buy with USDT.

Code Snippet

```
function buyWithUSDT(uint256 usdtAmount) external nonReentrant {
        require(usdtAmount > 0, "Zero USDT");
        require(
            usdt.transferFrom(msg.sender, treasury, usdtAmount),
            "USDT transfer failed"
        );

        uint256 usdAmount = usdtAmount * 1e12; // USDT has 6 decimals →
normalize to 1e18
        _processPurchase(msg.sender, usdAmount, "USDT");
    }
```

Recommendation

Use OpenZeppelin's SafeERC20.sol library.

Disclaimer

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identified.

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.