MoonPump

Audit Report

Wed Dec 25 2024

✉ contact@bitslab.xyz

https://twitter.com/scalebit_

ScaleBit

MoonPump Audit Report

1 Executive Summary

1.1 Project Information

Description	MoonPump is an AI-driven token launch platform that enables users to quickly create and launch MEMEcoins in real time, based on trending topics and discussions on popular social media platforms
Type	DeFi
Auditors	ScaleBit
Timeline	Tue Dec 17 2024 - Wed Dec 25 2024
Languages	Rust
Platform	Solana
Methods	Architecture Review, Unit Testing, Manual Review

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	11	11	0
Informational	1	1	0
Minor	3	3	0
Medium	5	5	0
Major	2	2	0
Critical	0	0	0

1.4 ScaleBit Audit Breakdown

ScaleBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow
- Number of rounding errors
- Unchecked External Call
- Unchecked CALL Return Values
- Functionality Checks
- Reentrancy
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Gas usage
- Fallback function usage
- tx.origin authentication
- Replay attacks
- Coding style issues

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** and **"Formal Verification"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by MoonPump to identify any potential issues and vulnerabilities in the source code of the MoonPump smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified 11 issues of varying severity, listed below.

ID	Title	Severity	Status
BCU-1	In the `swap()` Function, the Condition when `is_buy` is true is not Strict Enough	Medium	Fixed
BCU-2	The Calculation of `new_sol` should be Rounded Up	Medium	Fixed
BUY-1	Missing Slippage Protection for `amount_in`	Medium	Fixed
BUY-2	Insufficient Balance Check Does Not Include Trade Fee	Minor	Fixed
BUY-3	Missing Check for `trade_fee > 0`	Minor	Fixed
ERR-1	Unused Error Codes	Informational	Fixed
GRA-1	`bonding_curve_token_account` Lacks Constraints	Minor	Fixed
INI-1	Incorrect Space Allocation for Account Initialization	Medium	Fixed
SEL-1	The Sell Rent Logic is Incorrect	Major	Fixed

SEL-2	The Slippage Protection in the Sell Instruction is Incorrect	Major	Fixed
TOW-1	Incorrect Implementation of Two-Step Ownership Transfer	Medium	Fixed