QuillAudits

Audit Report
September, 2024

For

MONWU

Table of Content

Executive Summary

Project Name

MONWU

Project URL

http://monwu.com

Overview

MONWU contracts encompasses an ERC20 token creation designed to distribute these tokens to different department vesting contract addresses in accordance to their allocations. The token has a maximum cap of 1,000,000,000 and allows token holders to burn their tokens until it reaches its minimal cap of 500,000,000.

Both the Development and Marketing vesting contracts share similar features which allows vesting to last for 2 years and the releasable tokens will be half its total allocation for each year. While founders and private sale have a one year cliff period where allocated tokens get locked until cliff ends and vesting follows. Releasable tokens are alloted to 5 phases. Unclaimed private investors tokens are given a grace of one year after vesting period before the owner burns the tokens.

Audit Scope

The scope of this audit was to analyze the Monwu codebase for quality, security, and correctness.

Contracts In-Scope

MONWU Token : https://polygonscan.com/address/0xff0357cB7b46aC5BBf79B8cAF4FfbD02B87bfC3c

Founders : https://polygonscan.com/address/0x8b011a5bCb85E5c664981c07e69689B1C0092149

Marketing : https://polygonscan.com/address/0x474BC6E525AB4315551Fc4b0B6b5C260E8F60CFC

Development : https://polygonscan.com/address/0x5aE864994e71B3588306161ea3a224b279D19E16

Private Sale : https://polygonscan.com/address/0x3549E81DA41f453a6fE2332B71bCa4dc1E59C9EB

MonwuPrivateSaleWhitelist : https://polygonscan.com/address/0x353b6EDa579aa1AaCF7b603865919c93d507255B#code

Executive Summary

Commit Hash	NA
Language	Solidity
Blockchain	Polygon
Method	Manual Analysis, Functional Testing, Automated Testing
First Review	16th september 2024 - 18th September 2024
Updated Code Received	NA
Second Review	NA
Fixed In	NA

Number of Security Issues per Severity

0
Issues Found
🟥 High
🟨 Medium
🟩 Low
🟪 Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	0	0
Partially Resolved Issues	0	0	0	0
Resolved Issues	0	0	0	0

Checked Vulnerabilities

- ✓ Re-entrancy
- ✓ Arbitrary write to storage
- ✓ Centralization of control
- ✓ Ether theft
- ✓ Improper or missing events
- ✓ Logical issues and flaws
- ✓ Arithmetic Correctness
- ✓ Race conditions/front running
- ✓ SWC Registry
- ✓ Re-entrancy
- ✓ Timestamp Dependence
- ✓ Gas Limit and Loops
- ✓ Exception Disorder
- ✓ Gasless Send
- ✓ Use of tx.origin
- ✓ Malicious libraries
- ✓ Compiler version not fixed
- ✓ Address hardcoded
- ✓ Divide before multiply
- ✓ Integer overflow/underflow
- ✓ ERC's conformance
- ✓ Dangerous strict equalities
- ✓ Tautology or contradiction
- ✓ Return values of low-level calls
- ✓ Missing Zero Address Validation
- ✓ Private modifier
- ✓ Revert/require functions
- ✓ Multiple Sends
- ✓ Using suicide
- ✓ Using delegatecall
- ✓ Upgradeable safety
- ✓ Using throw

Checked Vulnerabilities

- ✓ Using inline assembly
- ✓ Style guide violation
- ✓ Unsafe type inference
- ✓ Implicit visibility level

Techniques and Methods

Throughout the audit of smart contracts, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments, match logic and expected behavior.
- Token distribution and calculations are as per the intended behavior mentioned in the whitepaper.
- Implementation of ERC standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods, and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behavior of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Foundry, Solhint, Mythril, Slither, Solidity statistic analysis.

Types of Severity

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

High Severity Issues

No issues were found.

Medium Severity Issues

No issues were found.

Low Severity Issues

No issues were found.

Informational Issues

No issues were found.

Functional Tests Cases

- ✓ Should create tokens, transfer ownership to the contract owner, and make distribution of allocated tokens possible
- ✓ Should allow only whitelist owner add, edit and remove addresses to the whitelist
- ✓ Should allow investors to claim their allocations when malicious owner removes them during vesting period
- ✓ Should make releasable tokens for individual allocations phased into 5 unlocks
- ✓ Should allow releasable tokens greater than zero after the cliff period is over
- ✓ Should make investors withdraw their allocated tokens in fractions - 20% of each 5 unlocks
- ✓ Should allow the contract owner to burn the left over tokens not claimed by an investor when burn time reaches
- ✓ Should allow the contract owner to withdraw exact amount in the contract

Automated Tests

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of MONWU We performed our audit according to the procedure described above.

Code Looks Good ,No issues Found.

Disclaimer

QuillAudits Smart contract security audit provides services to help identify and mitigate potential security risks in MONWU. However, it is important to understand that no security audit can guarantee complete protection against all possible security threats. QuillAudits audit reports are based on the information provided to us at the time of the audit, and we cannot guarantee the accuracy or completeness of this information. Additionally, the security landscape is constantly evolving, and new security threats may emerge after the audit has been completed.

Therefore, it is recommended that multiple audits and bug bounty programs be conducted to ensure the ongoing security of MONWU. One audit is not enough to guarantee complete protection against all possible security threats. It is important to implement proper risk management strategies and stay vigilant in monitoring your smart contracts for potential security risks.

QuillAudits cannot be held liable for any security breaches or losses that may occur subsequent to and despite using our audit services. It is the responsibility of MONWU to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.

About QuillAudits

QuillAudits is a leading name in Web3 security, offering top-notch solutions to safeguard projects across DeFi, GameFi, NFT gaming, and all blockchain layers. With six years of expertise, we've secured over 1000 projects globally, averting over $30 billion in losses. Our specialists rigorously audit smart contracts and ensure DApp safety on major platforms like Ethereum, BSC, Arbitrum, Algorand, Tron, Polygon, Polkadot, Fantom, NEAR, Solana, and others, guaranteeing your project's security with cutting-edge practices.

1000+
Audits Completed

$30B
Secured

1M+
Lines of Code Audited

Follow Our Journey

Audit Report
September, 2024

For

MONWU

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com

audits@quillhash.com