QuillAudits

AUDIT REPORT

December, 2024

For

ARTEMIS

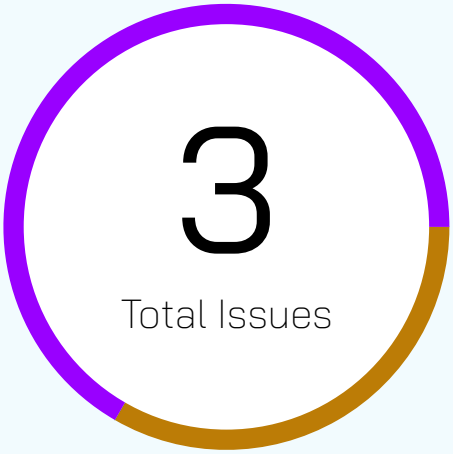
Table of Content

Executive Summary

Project name	Artemis Finance
Project URL	https://artemisfinance.io/home
Overview	It stakes Metis on behalf of users on Artemis. * It locks the received ArtMetis tokens and unlocks them only when a certain period (21 days) has passed. * The locking applies for each staking action. A user has to unlock multiple times if they did multiple * staking actions on different occasions. This is done using ERC1155Supply to keep track of staking action.
Audit Scope	The scope of this Audit was to analyze the Artemis Smart Contracts for quality, security, and correctness.
Contracts in Scope	StakeAndLock.sol BArtMetis.sol
Source Code link	https://drive.google.com/drive/folders/15e14dt-Fwkhhudg6E_9RxZeOdOY15yvi?usp=sharing
Language	Solidity
Blockchain	Metis
Methods	Manual Review, Functional Testing, Automated Testing, etc. All the raised flags were manually reviewed and re-tested to identify any false positives.
Review 1	16th December 2024 - 18th December 2024
Updated Code Received	20th December 2024

Review 2	23rd December 2024
Fixed in	582e51ba04361cb64ce4cbd0da578638b4891ba6

Number of Issues per Severity



High	0 (0.00%)
Medium	1 (33.33%)
Low	0 (0.00%)
Informational	2 (66.67%)

Severity

Issues	High	Medium	Low	Informational
Open	0	0	0	0
Resolved	0	0	0	**1**
Acknowledged	0	**1**	0	**1**
Partially Resolved	0	0	0	0

Checked Vulnerabilities

- ✓ Re-entrancy
- ✓ Timestamp Dependence
- ✓ Gas Limit and Loops
- ✓ DoS with Block Gas Limit
- ✓ Transaction-Ordering Dependence
- ✓ Use of tx.origin
- ✓ Exception Disorder
- ✓ Gasless Send
- ✓ Balance Equality
- ✓ Byte Array
- ✓ Transfer Forwards All Gas
- ✓ ERC20 API Violation
- ✓ Compiler Version Not Fixed
- ✓ Redundant Fallback Function
- ✓ Send Instead of Transfer
- ✓ Style Guide Violation
- ✓ Unchecked External Call
- ✓ Unchecked Math
- ✓ Unsafe Type Inference
- ✓ Implicit Visibility Level
- ✓ Access Management
- ✓ Arbitrary Write to Storage
- ✓ Ether Theft
- ✓ Centralization of Control
- ✓ Improper or Missing Events
- ✓ Logical Issues and Flaws
- ✓ Arithmetic Computations Correctness
- ✓ Race Conditions/Front Running
- ✓ Malicious Libraries
- ✓ SWC Registry
- ✓ 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
- ☑ Using Delegatecall
- ☑ Using Throw

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 statistical 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 Issues

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

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

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.

Medium Severity Issues

Centralization Risk

Acknowledged

Path

StakeandLock.sol

Function

emergencyRecoverToken

Description

The emergencyRecoverToken function allows the owner to withdraw funds at will. This introduces a significant centralization risk, as the owner could potentially misuse this function to execute a rug pull and drain the funds from the contract.

Recommendation

To mitigate the risk of misuse and enhance trust, it is recommended to transition critical decisions, such as fund recovery, to a decentralized governance mechanism.

Teams Comment

The owner of StakeandLock will be Metis multi-sig or Metis DAO contract.

Informational Severity Issues

Use strict pragma

Acknowledged

Path

BArtMetis.sol

Description

smart contract explicitly specifies an exact compiler version (e.g., pragma solidity 0.8.22;). This approach enhances security and reliability by ensuring consistency across deployments and preventing discrepancies caused by potential changes or bugs in different compiler versions. By locking the compiler version, the risk of incompatibility or unexpected behavior is mitigated, aligning with best practices for secure smart contract development.

Recommendation

Use pragma solidity ^0.8.0;

mintBatch and burnBatch functions can be omitted.

Resolved

Path

BArtMetis.sol

Function

mintBatch(), burnBatch()

Description

mintBatch and burnBatch functions are included in the contract but may be unnecessary for the intended use case. If batch operations are not critical, these functions increase the contract's complexity and gas overhead without adding significant value. Omitting them can simplify the contract, reduce the attack surface, and optimize gas usage during deployment and execution.

Recommendation

Remove mintBatch and burnBatch functions if batch processing is not essential

Functional Tests

Some of the tests performed are mentioned below:

- ✔ Users who deposit an amount must set _minArtMetisAmountToReceive param more than zero however here it is passed in case of zero.

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 the Artemis Finance contracts. We performed our audit according to the procedure described above.

One issues of Medium and informational severity were found, Some suggestions and best practices are also provided in order to improve the code quality and security posture.

Disclaimer

QuillAudits Smart contract security audit provides services to help identify and mitigate potential security risks in Artemis smart contract. 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 Artemis smart contract. 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 the Artemis 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.

6+ Years of Expertise	**1M+** Lines of Code Audited
$30B+ Secured in Digital Assets	**1K+** Projects Secured

Follow Our Journey

AUDIT REPORT

December, 2024

For

Canada, India, Singapore, UAE, UK

www.quillaudits.com audits@quillaudits.com