Nemo Protocol-ll

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Tue Jan 21 2025

Nemo Protocol-ll Audit Report

1 Executive Summary

1.1 Project Information

Description	Nemo Protocol is a decentralized finance application.
Type	DeFi
Auditors	MoveBit
Timeline	Fri Jan 17 2025 - Mon Jan 20 2025
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/nemo-protocol/nemo
Commits	5c64435fd99fb12a078e4738e62a5c38b0d7855d fa7fae52b59733a526ca15faa4af2c59b4f0a4c4 c2ade08409797e6c8bdb29577965ce6f18906404 957dfdb266ab3a6da6ee62d0bacfd3eb6af13809 34b7544bca6788b036fef7afb127af95067d9ed2 957dfdb266ab3a6da6ee62d0bacfd3eb6af13809

1.2 Files in Scope

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

ID	File	SHA-1 Hash
ACL	nemo_farming/sources/acl.move	8399c3096a1ae924d691dd859d5772bb509a705d
BBA	nemo_farming/sources/balance_bag.move	340e874a070d0f8b74c07bfa88c3ea2562e73dea
CON	nemo_farming/sources/config.move	3860e54ae068f9fca08290922d0dd9e6ad6e2b8d
POO	nemo_farming/sources/pool.move	76df69a8b780c234e985dd8ba5e70b13cc3f49d8
ERR	nemo_farming/sources/error.move	902968d4fa472e933530c69ce1f49737cfb6f5a4

1.3 Issue Statistic

Item	Count	Fixed	Partially Fixed	Acknowledged
Total	11	9	1	1
Informational	1	1	0	0
Minor	3	3	0	0
Medium	1	1	0	0
Major	4	2	1	1
Critical	2	2	0	0

1.4 MoveBit Audit Breakdown

MoveBit 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 by bit operations
- Number of rounding errors
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting
- Unchecked CALL Return Values
- The flow of capability
- Witness Type

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) Formal Verification(Optional)

Perform formal verification for key functions with the Move Prover.

(4) 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 Nemo to identify any potential issues and vulnerabilities in the source code of the Nemo Protocol-ll 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
CON-1	Lack of Version Check	Medium	Fixed
ERR-1	Unused Function	Informational	Fixed
POO-1	Type Mismatch and Validation Missing in Reward and Stake Operations	Critical	Fixed
POO-2	Missing Zero Stake Check in `update_acc_per_share`	Critical	Fixed
POO-3	Missing Emission Rate Validation	Major	Fixed
POO-4	Centralization Risk	Major	Acknowledged
POO-5	Redundant Assignment of `rewarder.active`	Minor	Fixed
REW-1	Lack of Method to Reduce `total_stake_shares`	Major	Partially Fixed
REW-2	Potential Unreachable Condition	Major	Fixed
CON1-1	Missing Version Upgrade Interface	Minor	Fixed

CON1-2	Lack of Events Emit	Minor	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the Nemo Protocol-ll Smart Contract :

Owner

- **Owner can** call the `add_acceleration_factor` function to configure the acceleration factor for specific pools.
- **Owner can** call the `register_pool` function to register a new pool in the system.
- **Owner can** call the `add_reward_pool` function to add a reward pool to the system.
- **Owner can** call the `enable_pool_reward` function to enable reward distribution for a specific pool.
- **Owner can** call the `emergent_stop_pool_rewarder` function to stop the reward distribution for a pool in emergency situations.
- **Owner can** call the `emergent_withdraw_reward` function to withdraw rewards from a pool in emergency situations.
- **Owner can** call the `remove_reward_pool` function to remove a reward pool from the system.
- **Owner can** call the `update_reward_pool_emission_rate` function to modify the emission rate of a reward pool.
- **Owner can** call the `add_reward_pool_total_reward` function to add more rewards to the total reward pool.

User

- **User can** call the `stake` function to stake their position into a pool and participate in rewards.
- **User can** call the `claim` function to claim their accumulated rewards from a pool.
- **User can** call the `unstake` function to withdraw their staked position from a pool.

4 Findings

CON-1 Lack of Version Check

Severity: Medium

Status: Fixed

Code Location:

nemo/vendor/haedal/sources/config.move#42

Descriptions:

In the `add_acceleration_factor`, `update_pool_rewarder_acc_per_share` functions, version checking is missing. This could allow operations between different versions to be compatible, potentially leading to unknown security risks.

Suggestion:

It is recommended to add a version check for the function.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

ERR-1 Unused Function

Severity: Informational

Status: Fixed

Code Location:

nemo_farming/sources/error.move#62,38-46

Descriptions:

There are unused functions in the contract, and in order to follow the code standard, it is recommended to delete the unused functions if there is no demand to use them in the future.

Suggestion:

It is recommended to remove the unused function if there's no further design.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

POO-1 Type Mismatch and Validation Missing in Reward and Stake Operations

Severity: Critical

Status: Fixed

Code Location:

nemo_farming/sources/pool.move#371 338

Descriptions:

The `add_reward_pool_total_reward` function does not validate if the type of the reward token being added matches the `reward_token` type for the pool rewarder. Additionally, the `stake` function lacks validation to ensure that the `MarketPosition` 's associated market is of the correct market type.

Suggestion:

It is recommended to enforce type checks in both functions.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

POO-2 Missing Zero Stake Check in `update_acc_per_share`

Severity: Critical

Status: Fixed

Code Location:

nemo_farming/sources/pool.move#612

Descriptions:

The `update_acc_per_share` function does not check whether `total_stake_shares` is zero. If it is zero, the calculation of `acc_per_share` will fail or result in an unintended division by zero scenario.

Suggestion:

It is recommended to add a check for `total_stake_shares` being zero. If it is zero, skip the update to `acc_per_share` and return early.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

POO-3 Missing Emission Rate Validation

Severity: Major

Status: Fixed

Code Location:

nemo_farming/sources/pool.move#167

Descriptions:

The `add_reward_pool` function does not verify whether `emission_per_second * (end_time - current_time)` is less than or equal to `total_reward`. This could result in a situation where the pool's total reward is insufficient, causing users to fail when attempting to withdraw their rewards.

Suggestion:

It is recommended to validate that `emission_per_second * (end_time - current_time) <= total_reward` during the addition of a reward pool to ensure sufficient funds are allocated for the specified emission rate and duration.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

POO-4 Centralization Risk

Severity: Major

Status: Acknowledged

Code Location:

nemo_farming/sources/pool.move#243

Descriptions:

Centralization risk was identified in the smart contract:

- Administrators can call `emergent_withdraw_reward()` to withdraw all the pool coins.

Suggestion:

It is recommended that measures be taken to reduce the risk of centralization, such as a multi-signature mechanism.

POO-5 Redundant Assignment of `rewarder.active`

Severity: Minor

Status: Fixed

Code Location:

nemo_farming/sources/pool.move#260

Descriptions:

The `emergent_withdraw_reward` function redundantly sets `rewarder.active = false` twice.

Suggestion:

It is recommended to remove the duplicate assignment of `rewarder.active = false` to improve code clarity and efficiency.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

REW-1 Lack of Method to Reduce `total_stake_shares`

Severity: Major

Status: Partially Fixed

Code Location:

nemo/sources/farming/reward.move

Descriptions:

`total_stake_shares` is used for calculating reward distribution. However, during the audit, we only found places where `total_stake_shares` was increased and didn't see any places where it was decreased. This may lead to an underestimation of the rewards during the reward calculation process.

Suggestion:

It is recommended to add a method to decrease the `total_stake_shares`.

Resolution:

The client added a `remove_stake_shares()` function to fix this issue. However, this new method has introduced a new issue. If a user removes their stake shares without first claiming their accrued rewards, those rewards become unclaimable.

REW-2 Potential Unreachable Condition

Severity: Major

Status: Fixed

Code Location:

nemo/sources/farming/reward.move

Descriptions:

The `remove_rewarder()` method is used to remove a rewarder. It first checks `total_reward` and `reward_harvested`. However, due to issues such as precision loss or user operations, this condition may not be met.

Suggestion:

It is recommended to modify the code to avoid the problem of being unable to remove the rewarder.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

CON1-1 Missing Version Upgrade Interface

Severity: Minor

Status: Fixed

Code Location:

nemo_farming/sources/config.move#26

Descriptions:

The module defines a `package_version` field for version control but lacks an interface to upgrade it. This restricts the protocol's upgradeability since there is no proper way to update the version number when needed.

Suggestion:

It is recommended to add an admin-only function to update the `package_version` field, with proper access control and event emission to track version changes.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

CON1-2 Lack of Events Emit

Severity: Minor

Status: Fixed

Code Location:

nemo_farming/sources/config.move#42

Descriptions:

The contract lacks appropriate events for some key functions. The lack of event records for these functions may cause inconvenience in the subsequent tracking and contract status changes.

Suggestion:

It is recommended that events for these functions be emitted.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.