Dola Protocol

Audit Report

contact@movebit.xyz

https://twitter.com/movebit_

Wed Feb 07 2024

Dola Protocol Audit Report

1 Executive Summary

1.1 Project Information

Description	DOLA protocol is a decentralized omnichain liquidity aggregation protocol with single coin pools of each public chain as the core, cross-chain messaging protocols such as Wormhole, Layerzero as the bridge, and Sui public chain as the settlement center
Type	DeFi
Auditors	MoveBit
Timeline	Mon Jan 15 2024 - Wed Feb 07 2024
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/OmniBTC/DolaProtocolDev
Commits	77b73587a8c2ee8251155c424f9c20d90869b718

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MOV25	sui/dola_protocol/Move.toml	1655cd50d8e0e7c3530f07540cfe7a6edd6cdc36
EFE	sui/dola_protocol/sources/pool_manager/equilibrium_fee.move	b75c4ea2b20b35be6750313810d3aa6a10c42ec8
PMA	sui/dola_protocol/sources/pool_manager/pool_manager.move	c2e090d758cb1e300e44a83ab7e08dc251f07ef4
ORA	sui/dola_protocol/sources/oracle/oracle.move	3ef9094a64961fe9e7e6642ed18ce3030bac6b26
MCO	sui/dola_protocol/sources/utils/merge_coins.move	1f258db59b90b235d5fd20a9015d99346d3a31f9
WAC	sui/dola_protocol/sources/wormhole_adapter_core/wormhole_adapter_core.move	3cb532978bfd0f7e69b60d90a5c09ab6a3ad2998
WAV1	sui/dola_protocol/sources/wormhole_adapter_core/wormhole_adapter_verify.move	85f11322827bd24fd1dafdbec32fcdd4d7014b42
RGC	sui/dola_protocol/sources/wormhole_adapter_core/remote_gov_codec.move	8478a83fd410fdbfe296244c687b0a8a48acfa10
WAD	sui/dola_protocol/sources/system_core/wormhole_adapter.move	428c868afbb9177d519c2e14fa4893624e015123
SCO1	sui/dola_protocol/sources/system_core/system_codec.move	d9aa5811c6d65dfc9ec2135c05201ab30f44560c

STO	sui/dola_protocol/sources/system_core/storage.move	7a8ab99c0af5adf6d4f1c14ef93826fb2f71756f
DAD1	sui/dola_protocol/sources/dola_types/dola_address.move	e5b2fe6c89dba45dc729727f25a0dda001e3d4e2
GEN	sui/dola_protocol/sources/governance/genesis.move	7c94470870738cf7013cb7dd6c908ae983882735
GV2	sui/dola_protocol/sources/governance/governance_v2.move	da9e350aa68710293d3486e96c1b6dfa88716eef
GV1	sui/dola_protocol/sources/governance/governance_v1.move	0d94412430f0cad59fd8feec202cb0283d5f88e6
LV2	sui/dola_protocol/sources/dola_portal/lending_v2.move	dc75aad42f8d6e2dd095516643c4f3b635c3decd
SYS1	sui/dola_protocol/sources/dola_portal/system.move	dbd213ffc603c413eafd6b6d13c240a4a3e5ce07
LEN1	sui/dola_protocol/sources/dola_portal/lending.move	8056610dbffc2dceeb58ab63aa75fec0bc47b30b
SV2	sui/dola_protocol/sources/dola_portal/system_v2.move	2fa6acf2f5498e27a4cf9384318f924463b18641
UMA	sui/dola_protocol/sources/user_manager/user_manager.move	60415b068c634b3c6228c6f8b87b7b2736bc1a5d
AMA	sui/dola_protocol/sources/app_manager/app_manager.move	9026e406c58437c3e7facb1ab6cb600911d50cf3
MAT	sui/dola_protocol/sources/ray_math/math.move	8473831999f4ae7da9eff9bc7eb9ee578fdab30c
BOO	sui/dola_protocol/sources/lending_core/boost.move	db21ae844fbc63301511eb8ea413e08f2e8bc44a

LTE	sui/dola_protocol/sources/lending_core/tests/logic_tests.move	c0876f9a8575062a9622c564a343f58405d1cd99
WAD1	sui/dola_protocol/sources/lending_core/wormhole_adapter.move	90833528c9bee235bfa5fea6f22d7159a0cb92e5
LOG	sui/dola_protocol/sources/lending_core/logic.move	d7d94a3a3847c95c83e4e1d7ef9b9c20aa26dd17
LCO1	sui/dola_protocol/sources/lending_core/lending_codec.move	2f34a14d5eb8c8b8376afc1a351e2fbcf06da2bd
STO1	sui/dola_protocol/sources/lending_core/storage.move	6fe45fa3182da1e9f310bc0bf832943f6595629b
RAT	sui/dola_protocol/sources/lending_core/rates.move	8f84498745a3849201b19c20a806c66d474478f7
SBA	sui/dola_protocol/sources/lending_core/scaled_balance.move	e4792f1b8df882d2b2b44bb3c60b8a26ec341d19
SER1	sui/dola_protocol/sources/serde/serde.move	eb92fdc588b690ceacb664f0b58aa85369898666
PCO1	sui/dola_protocol/sources/omnipool/pool_codec.move	e5bf7b03928b47c1443f840d636e45ea762a8732
WAP1	sui/dola_protocol/sources/omnipool/wormhole_adapter_pool.move	040e5621b0140afb65ec855debf3ec1854c3d2cb
DPO1	sui/dola_protocol/sources/omnipool/dola_pool.move	b96d62c068d86bfdf323e03fd3bd7582b6972270

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	9	7	2
Informational	1	1	0
Minor	3	3	0
Medium	1	1	0
Major	2	1	1
Critical	2	1	1

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

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 DOLA-Protocol to identify any potential issues and vulnerabilities in the source code of the Dola Protocol 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 9 issues of varying severity, listed below.

ID	Title	Severity	Status
BOO-1	New Users Can Get Rewarded Immediately	Critical	Fixed
BOO-2	Unnecessary `friend` Privileges	Minor	Fixed
DPO-1	`create_pool` Function Is Lack Of Permission Checking	Critical	Acknowledged
GV1-1	Lack of `UpgradeCap` id checking	Major	Acknowledged
GV2-1	`create_proposal` Has No Permission control	Major	Fixed
GV2-2	Incorrect Annotation	Informational	Fixed
LOG-1	Wrong Event Value	Medium	Fixed
LOG-2	Set the Deprecated Module `friend`	Minor	Fixed
PMA-1	Lack of Events Emit	Minor	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the Dola Protocol Smart Contract :

Admin

- The admin can register the cap through the `register_cap_with_governance()` function.
- The admin can destroy the app cap using the `destroy_app_cap()` function.
- The admin has the privilege to activate the current version of governance via the `activate_governance()` function.
- The admin can upgrade the governance using the `upgrade()` function.
- The admin can add guardians through the `setFreeWithdrawableRate()` function.
- The admin can remove the guardian by calling the `remove_guardians()` function.
- The admin can update minimum staking by calling the `update_minumum_staking()` function.
- The admin can invoke the `update_delay()` function to update the delay.
- The admin has the authority to `destroy_governance_cap()` functions to destroy governance cap.
- The admin has the ability to invoke the `register_pool_id()` function to create a new pool ID.
- The admin can utilize the `register_pool()` functions to add the pool of a certain chain to the dola pool ID.
- The admin has the capability to invoke the `set_pool_weight()` function to set the weight of the liquidity pool.
- The admin can set the alpha of the equilibrium fee through the `set_equilibrium_alpha()` function.
- The admin is able to set the lambda of equilibrium fee using the `set_equilibrium_lambda()` function.
- The admin has the privilege to register the chain IDs that need to be grouped via the `register_dola_chain_id()` function.

- The admin can unregister the chain IDs that need to be grouped using the `unregister_dola_chain_id()` function.
- The admin can initialize caps of PoolManager and UserManager by calling the `initialize_cap_with_governance()` function.
- The admin can register the remote wormhole adapter pool with the `register_remote_bridge()` function.
- The admin can delete the remote wormhole adapter pool by calling the `delete_remote_bridge()` function.
- The admin can invoke the `remote_register_spender()` function to register spender for remote bridge.
- The admin has the ability to invoke the `remote_delete_spender()` function to delete the spender for a remote bridge.
- The admin can utilize the `remote_add_relayer()` functions to add the relayer for the remote bridge.
- Users can remove the relayer for the remote bridge by calling the `remote_remove_relayer()` function.
- The admin can invoke the `add_relayer()` function to add relayer.
- The admin can invoke the `remove_relayer()` function to remove relayer.
- The admin can set the vaa expired time through the `set_vaa_expired_time()` function.
- The admin can invoke the `register_new_reserve()` function to register a ReserveData.
- The admin has the ability to invoke the `set_is_isolated_asset()` function to set whether this asset is isolated.
- The admin can utilize the `set_borrowable_in_isolation()` functions to set whether they can borrow in isolation.
- The admin can set the treasury factor by calling the `set_treasury_factor()` function.
- The admin can invoke the `set_supply_cap_ceiling()` function to set supply celling.
- The admin can invoke the `set_borrow_cap_ceiling()` function to set borrow cap celling.
- The admin can set the collateral coefficient through the `set_collateral_coefficient()` function.

- The admin is able to set the borrow coefficient using the `set_borrow_coefficient()` function.
- The admin has the privilege to set borrow rate factors via the `set_borrow_rate_factors()` function.
- The admin can create shadow coins using the `create_and_init_boost_coins()` function.
- The admin can create a reward pool with boost coin by the `create_reward_pool_with_boost_coin()` function.

Relayer

- Relayer have the ability to call the `receive_withdraw()` function to receive withdraw.

User

- Users can use the `create_proposal()` and `create_proposal_with_history()` functions to create the proposal.
- Users can invoke the `vote_proposal()` function to vote for a proposal.
- Users can use the `cancel_proposal()` function to cancel the proposal.
- Users can call the `supply()` function to execute supply.
- Users can call the `withdraw()` function to withdraw the user`s token.
- Users have the ability to call the `borrow()` function to borrow.
- Users have the ability to call the `repay()` function to pay off debts.
- Users have the ability to call the `liquidate()` function to perform liquidation.
- Users can invoke the `as_collateral()` function to set assets as collateral.
- Users can utilize the `cancel_as_collateral()` function to unset assets as collateral.
- Users have the option to call the `sponsor()` function to mint the boost coin.
- Users can use the `claim_reward()` functions to claim the reward.
- Users can invoke the `bind_user_address()` function to bind user address.
- Users can use the `unbind_user_address()` function to unbind user address.

4 Findings

BOO-1 New Users Can Get Rewarded Immediately

Severity: Critical

Status: Fixed

Code Location:

sui/dola_protocol/sources/lending_core/boost.move#506

Descriptions:

Users get rewards through the `claim_reward` function, which will call the `update_user_reward` to update the rewards of user.Then `update_user_reward` calculate the `delta_index` between the user's `last_update_reward_index` and the `reward_index` recorded in the pool. However, for a new user, his `index_rewards_paid` is 0 by default, which means that he can get rewards in the range `reward_index` and 0 directly, causing a loss of rewards in the protocol as result.

Suggestion:

It is recommended to use `pool.reward_index` to initialize the `last_update_reward_index` for new users, or to restrict users from calling lending contracts directly.

BOO-2 Unnecessary `friend` Privileges

Severity: Minor

Status: Fixed

Code Location:

sui/dola_protocol/sources/lending_core/boost.move#607

Descriptions:

Some functions in the `boots.move` module have unnecessary `friend` privileges, such as `withdraw_boost_coin`, `mint_boost_coin`, `burn_boost_coin`, etc., which are only used in this module, but are still given `friend` privileges, so it's recommended to remove them.

Suggestion:

It is recommended to remove the unnecessary permissions.

DPO-1 `create_pool` Function Is Lack Of Permission Checking

Severity: Critical

Status: Acknowledged

Code Location:

aptos/omnipool/sources/dola_pool.move#82

Descriptions:

The lack of permission checks on the `create_pool` function allows anyone to call the function to create a pool, which is inconsistent with the design of the protocol. At the same time, `convert_pool_to_dola` converts the pool's `CoinType` to `dola_address`, which can allow users to create the same pool at will, which can result in a fake deposit or withdraw message being delivered by the bridge.

Suggestion:

It is suggested to add permission control to `create_pool` function

GV1-1 Lack of `UpgradeCap` id checking

Severity: Major

Status: Acknowledged

Code Location:

sui/dola_protocol/sources/governance/governance_v1.move#158

Descriptions:

The `activate_governance` function does not do a check on the source and type of the `upgrade_cap`, resulting in the user being able to pass in any module's `UpgradeCap` object to activate the governance.

Suggestion:

It is recommended to check the id of `UpgradeCap` to make sure it is originating from this module.

GV2-1 `create_proposal` Has No Permission control

Severity: Major

Status: Fixed

Code Location:

sui/dola_protocol/sources/governance/governance_v2.move#342

Descriptions:

The `create_proposal` function has no permission control, resulting in any user being able to create a proposal and vote for their own proposal, and being able to get `GovernanceCap` after a successful call to `vote_proposal`, resulting in a malicious user being able to gain access to the protocol.

Suggestion:

It is recommended to add the relevant privilege control.

GV2-2 Incorrect Annotation

Severity: Informational

Status: Fixed

Code Location:

sui/dola_protocol/sources/governance/governance_v2.move#479-499

Descriptions:

In the annotation, the proposal can be canceled by the creator, while in the `cancel_proposal` function, the proposal can only be canceled by the `guardian`, if the proposal creator is added to the `guardian`, then he can also cancel any other proposal.

Suggestion:

It is recommended to check whether the code and annotations are correct, add `|| sender == proposal.creator;` code, or update the annotations.

LOG-1 Wrong Event Value

Severity: Medium

Status: Fixed

Code Location:

sui/dola_protocol/sources/lending_core/logic.move#116

Descriptions:

In the `emit_reserve_stats` function, the `supply_index` field incorrectly uses the return value of the `get_borrow_index` function, which is the same as the `borrow_index`, as the value of the event, which may cause confusion in analyzing the data off the chain. Also the `UpdateUserRewardEvent` event in the `claim_reward` function, `old_reward_index` and `new_reward_index` also use `user_reward.last_update_reward_index` as a value

Suggestion:

It is recommended to change to the correct event value.

LOG-2 Set the Deprecated Module `friend`

Severity: Minor

Status: Fixed

Code Location:

sui/dola_protocol/sources/lending_core/logic.move#21

Descriptions:

The `lending_logic` module sets up the deprecated `lending_portal` as a `friend` module, and also no functions from `lending_logic` are used in `lending_v2`.

Suggestion:

It is suggested to delete the meaningless statements.

PMA-1 Lack of Events Emit

Severity: Minor

Status: Fixed

Code Location:

sui/dola_protocol/sources/pool_manager/pool_manager.move#338-424

Descriptions:

The smart contract lacks appropriate events for monitoring sensitive operations, which could make it difficult to track sensitive actions or detect potential issues.

Suggestion:

It is recommended to emit events for those sensitive functions.

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.