CAPA Token Contracts Audit Report

Prepared for Capapult, 17th January 2023

Table of Contents

Introduction

SCV was engaged by Capapult to assist in identifying security threats and vulnerabilities that have the potential to affect their security posture. Additionally, SCV will assist the team in understanding the risks and identifying potential mitigations.

Scope

SCV performed the security assessment on the following codebase:

- https://github.com/capapult-finance/capa-token
- Code Freeze: *f3ac9c2648a5a7afe7538619c073bd8be8f9b9f8*

Remediations were applied by the Capapult team up to the following commit hash:

- Code Freeze: *45cccd907fabc78261ff0759f3de4b1f9ef22c21*

Methodologies

SCV performs a combination of automated and manual security testing based on the scope of testing. The testing performed is based on the extensive experience and knowledge of the auditor to provide the greatest coverage and value to Capapult. Testing includes, but is not limited to, the following:

- Understanding the application and its code base purpose;
- Deploying SCV in-house tooling to automate dependency analysis and static code review;
- Analyse each line of the code base and inspect application security perimeter;
- Review underlying infrastructure technologies and supply chain security posture;

Code Criteria and Test Coverage

This section below represents how *SUFFICIENT* or *NOT SUFFICIENT* each code criteria was during the assessment

Criteria	Status	Notes
Provided Documentation	**SUFFICIENT**	N/A
Code Coverage Test	**SUFFICIENT**	N/A
Code Readability	**SUFFICIENT**	N/A
Code Complexity	**SUFFICIENT**	N/A

Vulnerabilities Summary

#	Summary Title	Risk Impact	Status
1	execute_poll_messages can be exploited to execute malicious messages	**Medium**	**Resolved**
2	Incorrect vesting schedule calculation	**Medium**	**Resolved**
3	New vesting schedule will overwrite existing vesting schedule	**Low**	**Resolved**
4	Contracts should use two-step ownership transfer	**Low**	**Acknowledged**
5	Additional funds sent to lockdrop are lost	**Low**	**Resolved**
6	Replace magic numbers	**Low**	**Resolved**
7	quorum and threshold can be set to 0	**Low**	**Resolved**
8	quorum and threshold are not validated during update_config	**Low**	**Resolved**
9	Lockdrop missing time validations	**Low**	**Resolved**
10	Distribute function is permissionless	**Low**	**Resolved**
11	Genesis time should not be an updatable value	**Low**	**Resolved**
12	stake_voting_tokens emitting incorrect attributes	**Informational**	**Resolved**
13	Update misleading comment	**Informational**	**Resolved**
14	Incorrect error type used	**Informational**	**Resolved**
15	Missing address validation for recipient address	**Informational**	**Resolved**
16	Remove deprecated parameter	**Informational**	**Resolved**

Detailed Vulnerabilities

1 - `execute_poll_messages` can be exploited to execute malicious messages

Risk Impact: Medium - **Status**: Resolved

Description

The `execute_poll_messages` function in `contracts/gov/src/contract.rs:528` can be exploited by a malicious poll to execute the messages for another poll. While `execute_poll_messages` restricts the caller to be the contract, this condition could be met if the call is happening in the context of another poll's message execution. The function does not check that the `poll_id` passed by the caller is `Passed`.

For example, an attacker could create a benign looking Poll A, that has a legitimate proposal and in the execute messages which could include a `ExecutePollMsgs` message and specify the id of a malicious Poll B. When Poll A passes, the `ExecutePollMsgs` message will be passed and Poll B's messages will also be executed without the poll having to be `Passed`. This could have a catastrophic effect.

In addition, a poll can also call its own messages multiple times because there is no check being performed to ensure the poll's messages have not already been executed.

Recommendations

We recommend ensuring that `execute_poll_messages` can only be executed once for a specific poll.

2 – Incorrect vesting schedule calculation

Risk Impact: Medium - **Status**: Resolved

Description

In the `register_vesting_accounts` function in `contracts/vesting/src/contract.rs:117` the vesting info's `last_claim_time` is set to `config.genesis_time`. This will result in an incorrect first claim as the account will be able claim for a time period in which they did not have an active vesting account.

Recommendations

We recommend setting `last_claim_time` to the time at which the vesting account is registered in `register_vesting_accounts`.

3 – New vesting schedule will overwrite existing vesting schedule

Risk Impact: Low - **Status**: Resolved

Description

In the `register_vesting_accounts` function in `contracts/vesting/src/contract.rs:104`, there is no validation to check that a vesting schedule does not exist for an address. If there is an existing vesting schedule for an address, it will simply be overwritten silently.

Recommendations

We recommend raising an error if a vesting schedule is provided for an address with an existing vesting schedule.

4 – Contracts should use two–step ownership transfer

Risk Impact: Low - **Status**: Acknowledged

Description

The current ownership transfer for each of the contracts is executed in one step, which imposes a risk that if the new owner is incorrect, then the admin privileges of the contract are effectively transferred and lost. A two-step ownership transfer is best practice because it requires the new admin to accept ownership before the transfer and config changes occur.

- `contracts/gov/src/contract.rs:230`
- `contracts/vesting/src/contract.rs:75`
- `contracts/lockdrop/src/contract.rs:528`

Recommendations

We recommend implementing a two-step ownership transfer where the current owner proposes a new owner address, and then that new owner address must call the contract to accept ownership within a finite time frame.

5 – Additional funds sent to lockdrop are lost

Risk Impact: Low - **Status**: Resolved

Description

The `deposit` function in `contracts/lockdrop/src/contract.rs:178` does not check to ensure that additional funds are not sent with the deposit message. This will result in any funds that are not luna to become lost.

Recommendations

We recommend validating that no additional funds are sent with the deposit message.

6 – Replace magic numbers

Risk Impact: Low - **Status**: Resolved

Description

Throughout the codebase, magic numbers are used. Magic numbers are hard-coded numbers without context. The use of magic numbers reduces the readability and maintainability of the codebase

Instances of magic numbers are listed below:

- `contracts/vesting/src/contract.rs:127`
- `contracts/lockdrop/src/contract.rs:373`
- `contracts/lockdrop/src/contract.rs:241`
- `contracts/lockdrop/src/contract.rs:437`

Recommendations

We recommend replacing the magic numbers mentioned above with a constant that is descriptive of its value and use case.

7 – `quorum` and `threshold` can be set to `0`

Risk Impact: Low - **Status**: Resolved

Description

The `validate_quorum` and `validate_threshold` functions in `contracts/gov/src/contract.rs:302-312` do not validate that the values are not `0`. These variables should represent meaningful parameters to ensure that substantial votes and participation are required to pass a poll.

Recommendations

We recommend adding a condition to ensure that the `validate_quorum` and `validate_threshold` are not `0`.

8 – `quorum` and `threshold` are not validated during `update_config`

Risk Impact: Low - **Status**: Resolved

Description

In the `update_config` function in `contracts/gov/src/contract.rs:212` the values of `quorum` and `threshold` are not properly validated in the same manner that they are validated during the contract's instantiation.

Recommendations

We recommend validating these values with the `validate_quorum` and `validate_threshold` functions.

9 – Lockdrop missing time validations

Risk Impact: Low - **Status**: Resolved

Description

In the `update_config` function in `contracts/lockdrop/src/contract.rs:549` and `553`, the new values set to `end_deposits_at` and `start_withdraws_from` have no logic validations with each other. For example, when updating `end_deposits_at`, the final value should be validated to be lower or equal to `start_withdraws_from`. Without this validation, funds submitted by users after a lockdrop starts would be not retrievable.

In the `instantiate` function in `contracts/lockdrop/src/contract.rs:40-42`, the values of `start_deposits_at`, `end_deposits_at` and `start_withdraws_from` are also missing the same time validations as mentioned above.

Recommendations

We recommended adding validation checks to parameters that require this logical validation. For example, the value of `start_deposits_at` should be validated to be lower than the value of `end_deposits_at` that is set and vice versa. The same applies to the values of `end_deposits_at` and `start_withdraws_from`.

10 – `Distribute` function is permissionless

Risk Impact: Low - **Status**: Resolved

Description

The `distribute` function in `contracts/collector/src/contract.rs:78` does not check the caller address. This would mean that anyone can execute the `distribute` function in the collector contract, albeit the comments mentioning that only the contract itself can execute the distribute function.

Recommendations

We recommend validating the caller address to only allow privilege addresses to execute the function. If this is intended, we recommend updating the comments to reflect this.

11 – Genesis time should not be an updatable value

Risk Impact: Low - **Status**: Resolved

Description

In the `update_config` function in `contracts/vesting/src/contract.rs:84`, the value of `genesis_time` can be updated after the contract's instantiation. Changing the `genesis_time` could introduce unintended consequences for existing vesting accounts.

Recommendations

We recommend removing the ability to update the value of `genesis_time` to prevent unintended behaviours from occurring unless there is a specific reason for allowing the genesis to be updatable, in this case we recommend documenting the specific circumstances that require the parameter to be updatable.

12 – `stake_voting_tokens` emitting incorrect attributes

Risk Impact: Informational - **Status**: Resolved

Description

The `stake_voting_tokens` function in `contracts/gov/src/staking.rs:65` is incorrectly emitting the `amount` value as the `share` attribute. This may cause issues when the attributes are indexed by block explorers or other tools.

Recommendations

We recommend updating the `share` attribute in `contracts/gov/src/staking.rs:65`.

13 – Update misleading comment

Risk Impact: Informational - **Status**: Resolved

Description

The `update_config` function in `contracts/vesting/src/contract.rs:68` contains a comment that specifies *"everyone can execute update config"* , which is incorrect. This function is only able to be executed by the contract's owner.

In addition, the `spend` function in `contracts/community/src/contract.rs:68` contains a comment mentioning *"ANC token"* when it should be *"CAPA token"* instead.

Recommendations

We recommend updating the comment to better reflect the function's functionality.

14 – Incorrect error type used

Risk Impact: Informational - **Status**: Resolved

Description

The `withdraw` function in `contracts/lockdrop/src/contract.rs:278` and `282` returns an incorrect error message. The errors are currently returning a `InvalidWithdrawTime` error.

Recommendations

We recommend updating the error message to better reflect the specific specific error being raised

15 – Missing address validation for recipient address

Risk Impact: Informational - **Status**: Resolved

Description

In the `spend` function in `contracts/community/src/contract.rs:90` there are no checks to ensure that the recipient address is a valid address before executing the funds transfer.

Recommendations

We recommend adding a check to ensure that the recipient is a valid address before executing the transfer.

16 – Remove deprecated parameter

Risk Impact: Informational - **Status**: Resolved

Description

In the `instantiate` function in `contracts/gov/src/contract.rs:55` and in `contracts/gov/src/state.rs:31` there is an unused deprecated parameter `expiration_period`. It is best practice to remove deprecated code.

Recommendations

We recommend removing these deprecated parameters.

Document control

Version	Date	Approved by	Changes
0.1	12/01/2023	Vinicius Marino	Document Pre-Release
0.2	16/01/2023	SCV Team	Remediation Revisions
1.0	17/01/2023	Vinicius Marino	Document Release

Appendices

A. Appendix – Risk assessment methodology

A qualitative risk assessment is performed on each vulnerability to determine the impact and likelihood of each.

Risk rate will be calculated on a scale. As per criteria Likelihood vs Impact table below:

	Rare	Unlikely	Possible	Likely
Critical	**Medium**	**Severe**	**Critical**	**Critical**
Severe	**Low**	**Medium**	**Severe**	**Severe**
Moderate	**Low**	**Medium**	**Medium**	**Severe**
Low	**Low**	**Low**	**Low**	**Medium**
Informational	**Informational**	**Informational**	**Informational**	**Informational**

LIKELIHOOD

- Likely: likely a security incident will occur;
- Possible: It is possible a security incident can occur;
- Unlikely: Low probability a security incident will occur;
- Rare: In rare situations, a security incident can occur;

IMPACT

- Critical: May cause a significant and critical impact;
- Severe: May cause a severe impact;
- Moderate: May cause a moderated impact;
- Low: May cause low or none impact;
- Informational: May cause very low impact or none.

B. Appendix – Report Disclaimer

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts SCV-Security to perform a security review. The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only. The content of this audit report is provided "as is", without representations and warranties of any kind, and SCV-Security disclaims any liability for damage arising out of, or in connection with, this audit report.

Copyright of this report remains with SCV-Security.