TFM
P2P-Contract
Audit Report

Prepared for TFM, 20th February 2023

Table of Contents

Introduction

SCV was engaged by TFM 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/tfm-com/audit_p2p
- Code Freeze: `43fb8625b464b75a8a8706527f156bc9754ce8dc`

Remediations were applied by the team and reviewed by SCV up to the hash:

- Code Freeze: `f408cec62cb5b61a8ffb4a2ad33c1599c23330e3`

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 FTM. 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**	69.90% coverage 418/598 line
Code Readability	**SUFFICIENT**	The codebase had good readability and utilised many Rust and CosmWasm best practices.
Code Complexity	**SUFFICIENT**	N/A

Vulnerabilities Summary

#	Summary Title	Risk Impact	Status
1	`Ask` asset's token address is not validated	**Low**	**Resolved**
2	Fees are deducted from the original order amount, potentially confusing query callers	**Low**	**Resolved**
3	`query_simulate` for past orders might return incorrect information	**Low**	**Resolved**
4	`query_config` does not return `proposed_owner` config variable	**Low**	**Resolved**
5	Successful contract instantiation does not emit attributes or events	**Informational**	**Resolved**
6	Cancel order does not emit asset information	**Informational**	**Resolved**
7	Incorrect `idx_namespace` used for `maker_taker_addr`	**Informational**	**Resolved**
8	Remove unused code	**Informational**	**Resolved**
9	Misleading action emitted for `accept_ownership` and `withdraw_fees` function	**Informational**	**Resolved**

Detailed Vulnerabilities

1 – `Ask` asset's token address is not validated

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

Description

When submitting an order in `contracts/p2p/src/order.rs:13`, the `ask_asset` may be a native token or a `CW20` token address. If the `ask_asset` is a `CW20` token address, it is not validated as a valid address.

Recommendations

Consider validating the address using `ask_asset.info.check(deps.api).unwrap()`.

2 – Fees are deducted from the original order amount, potentially confusing query callers

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

Description

In `contracts/p2p/src/order.rs:157-158`, the order's `offer` and `ask` asset is deducted by the maker and taker fee accordingly. This causes the order to lose track of its original `offer` and `ask` asset value after a successful trade.

Suppose a user calls `query_order` to track past orders. The user cannot differentiate whether the `offer_amount` and `ask_amount` are the original amount submitted or the amount after fees. This means the user cannot determine whether the order is successful or canceled other than querying past transactions.

Additionally, the `order_open` boolean cannot distinguish between successful and canceled trades because both would set the variable value to false.

Recommendations

Instead of deducting the fees directly from `order.offer_asset.amount` and `order.ask_asset.amount`, consider creating a temporary variable that holds the deducted value to preserve the original amount.

Additionally, the following variables can be introduced in the `OrderInfo` struct to distinguish between successful and canceled trades:

- `maker_fee: Option<Uint128>` to record the maker fee calculated in line `154`.
- `taker_fee: Option<Uint128>` to record the taker fee calculated in line `155`.

If both `maker_fee` and `taker_fee` are `Some(Uint128)`, users can conclude that this is a successful order. On the other side, users can conclude a canceled order if both values are `None`.

This fixes the issue, as the original `offer` and `ask` amount are still preserved while allowing users to query the maker and taker fee amount.

3 – `query_simulate` for past orders might return incorrect information

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

Description

In `contracts/p2p/src/query.rs:119`, the `query_simulate` function fetches the order's `offer` and `ask` asset amount to simulate the amount that will be received by the maker and taker based on the current configured fees.

This is problematic for past orders because the `offer` and `ask` asset amount is not initial as it includes the deducted fees as seen in `contracts/p2p/src/order.rs:157-158`. As a result, the `query_simulate` function will deduct the fees again from both assets' amounts, which is incorrect.

Besides that, the contract owner might increase or decrease the fees after a successful trade. This will cause the `maker_receives` and `taker_receives` to return incorrect values as the intended amount should be higher or lower.

Recommendations

Consider only allowing simulating open orders to prevent confusion.

4 - `query_config` does not return `proposed_owner` config variable

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

Description

When calling the `query_config` function in `contracts/p2p/src/query.rs:15`, the response does not include `proposed_owner`. This implies that no one can query the value of `proposed_owner` other than checking past transactions.

Recommendations

Consider including the `proposed_owner` variable in the `ConfigResponse` struct.

5 – Successful contract instantiation does not emit attributes or events

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

Description

When a contract is instantiated, a lack of emitted attributes or events in a successful instantiation on `contracts/p2p/src/contract.rs:47` hinders off-chain listeners from indexing the parameters configured by the contract owner.

Recommendations

Consider emitting relevant events or attributes based on configured parameters.

6 – Cancel order does not emit asset information

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

Description

In `contracts/p2p/src/order.rs:120`, canceling the order results in the emitted amount of the `offer` asset being returned. However, the asset information of the `offer` asset is not emitted, requiring off-chain listeners to query the specific order to retrieve the asset information.

Recommendations

Consider emitting the `offer` asset information along with the amount.

7 – Incorrect `idx_namespace` used for `maker_taker_addr`

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

Description

In `contracts/p2p/src/state.rs:50`, the `idx_namespace` used for `maker_taker_addr` is "`orders__taker_maker`." This is incorrect as the order is `maker` first, then the `taker`.

Recommendations

Consider modifying the `idx_namespace` to be "`orders__maker_taker`."

8 – Remove unused code

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

Description

In `contracts/p2p/src/state.rs:56-57`, instances of unused code are found. This impacts the readability of the codebase.

Recommendations

Consider removing the unused code.

9 – Misleading action emitted for `accept_ownership` and `withdraw_fees` function

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

Description

In `contracts/p2p/src/utils.rs:59` and `231`, the value of the "*action*" attribute key emitted is "*change_owner*" for the `accept_ownership` function and "*withdraw_address*" for the `withdraw_fees` function. This is misleading because other emitted actions follow the executed function name.

Recommendations

Consider modifying the values to the associated function name, i.e., modifying the attribute value to "*accept_ownership*" in line `59` and "*withdraw_fees*" in line `231`.

Document control

Version	Date	Approved by	Changes
0.1	17/02/2023	SCV-Security Team	Document Pre-Release
0.2	19/02/2023	SCV-Security Team	Remediation Revisions
1.0	20/02/2023	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.