AUDIT REPORT

Neutron

Airdrop Transfer

Prepared by SCV-Security

On 18th November 2023

Table of Contents

Introduction

SCV has been engaged by Neutron to conduct a comprehensive security review with the goal of identifying potential security threats and vulnerabilities within the codebase. The purpose of this audit is to evaluate the security posture of the codebase and provide actionable recommendations to mitigate any identified risks. This report presents an overview of the findings from our security audit, outlining areas of concern and proposing effective measures to enhance the codebase's security.

Scope Functionality

The contract is designed to facilitate the transfer of NTRN tokens to the Cosmos Hub Community Pool, adhering to specific requirements. All handlers within the contract must operate on a permissionless basis. The initial handler's responsibility is to create an IBC connection on the Hub.

Submitted Codebase

Airdrop Transfer Contract	
Repository	https://github.com/neutron-org/neutron-airdrop-transfer
Commit	`be865cff4aae882724cf3f901542116f630287b6`
Branch	`main`

Revision Codebase

Airdrop Transfer Contract	
Repository	https://github.com/neutron-org/neutron-airdrop-transfer/pull/4
Commit	`6b4c0f3029d17b36076d066d23ba6e1023054a09`
Branch	`main`

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 Neutron. Testing includes, but is not limited to, the following:

- Understanding the application and its functionality 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

This section provides an evaluation of specific criteria aspects as described below:

- **Documentation:** Evaluating the presence and comprehensiveness of publicly available or provided explanatory information, diagram flowcharts, comments, and supporting documents to enhance code understanding.
- **Coverage:** Evaluating whether the code adequately addresses all necessary cases and scenarios, ensuring that the intended functionality or requirements are sufficiently covered.
- **Readability:** Assessing how easily the code can be understood and maintained, considering factors such as code structure, naming conventions, and overall organization.
- **Complexity:** Evaluating the complexity of the code, including factors such as, number of lines, conditional statements, and nested structures.

The status of each criteria is categorized as either **SUFFICIENT** or **NOT-SUFFICIENT** based on the audit assessment. This categorisation provides insights to identify areas that may require further attention and improvement.

Criteria	Status	Notes
Documentation	**SUFFICIENT**	The documentation provided by the client provides sufficient coverage of the codebase.
Coverage	**SUFFICIENT**	The codebase had sufficient testing including local chain tests that covered many advanced edge cases.
Readability	**SUFFICIENT**	Overall the code was readable, although the codebase lacked code comments.
Complexity	**SUFFICIENT**	The codebase complexity was kept a minimum by a clear and concise design.

Findings Summary

Summary Title	Risk Impact	Status
Poorly formed messages can cause errors and block legitimate interchain transactions	SEVERE	RESOLVED
Error in sudo_timeout will block interchain transactions	SEVERE	PARTIALLY RESOLVED
Unchecked interchain fee amount	INFO	ACKNOWLEDGED
ICA message handling can be improved	INFO	ACKNOWLEDGED
Remove commented code blocks	INFO	ACKNOWLEDGED
ibc_fee_from_funds ignores excess denoms	INFO	ACKNOWLEDGED
Avoid emitting default attributes	INFO	ACKNOWLEDGED

Findings Technical Details

1. Poorly formed messages can cause errors and block legitimate interchain transactions

RISK IMPACT: SEVERE	STATUS: RESOLVED

Revision Notes

The client has removed the `INTERCHAIN_TX_IN_PROGRESS` flag from the contract and has created a set amount that will be used to fund the community pool. This addresses the situation where a user can purposefully block interchain transactions by specifying an incorrect value. Given the expected use of the function, the client has decided that it is acceptable to specify a constant value rather than tracking or querying the balance of the ICA.

Description

The `execute_fund_community_pool` function in `src/contract.rs:158` allows any caller to trigger an interchain transaction to call `MsgFundCommunityPool` from the interchain account associated with the contract. Currently there is a condition that ensures that only one interchain transaction is in progress at a time. The function does not currently validate that the amount value specified by the caller is a valid value so that the interchain transaction will not fail.

An attacker or normal user can accidentally or purposefully specify either `0` or an amount that is greater than the current balance of the ICA, such that the interchain transaction will be dispatched but then will error on the destination chain. `execute_fund_community_pool` can be called with an invalid amount to block legitimate transactions because `INTERCHAIN_TX_IN_PROGRESS` will be set to true.

Recommendation

We recommend validating that the amount sent in `execute_fund_community_pool` is not `0` and is also less than or equal to the balance of the interchain account. This can be achieved in 2 main ways. The first option is to configure a state variable that tracks the amount of funds that have been transferred to the ICA, and update this value whenever additional funds are sent or whenever `execute_fund_community_pool` is called. Then the amount parameter can be removed from the message and the contract variable can directly specify a valid amount. The second option is to query the balance of the ICA, and either ensure the amount specified by the caller is valid, or to fund the entire balance of the ICA to the community pool. Additionally it could be helpful to consider adding an admin entrypoint that allows for the owner to override `INTERCHAIN_TX_IN_PROGRESS` to false if an unexpected error is encountered.

2. Error in `sudo_timeout` will block interchain transactions

RISK IMPACT: SEVERE	STATUS: PARTIALLY RESOLVED

Revision Notes

This issue has been marked as partially resolved. The client has removed the `INTERCHAIN_TX_IN_PROGRESS` flag from the contract, so if the aforementioned error occurs, the contract will not remain in a locked state.

Description

In the `sudo_timeout` function in `src/contract.rs:298` , `source_port` is unwrapped. In a situation where the value is `None` an error is returned. This can create a situation where the `sudo_timeout` function errors and the `INTERCHAIN_TX_IN_PROGRESS` does not get reset to false.

Recommendation

We recommend configuring the contract so the edge cases associated with handler logic such as `sudo_timeout` will not cause the contract to be stuck in an error intermediate state. To accomplish this, the function can either use `unwrap_or("".to_string())` instead of `ok_or_else`, or the contract can add an admin entrypoint where the admin can intervene during the unlikely chance state is encountered and manually set `INTERCHAIN_TX_IN_PROGRESS` to false.

3. Unchecked interchain fee amount

RISK IMPACT: INFO	STATUS: ACKNOWLEDGED

Description

In the `execute_send_claimed_tokens_to_ica` and `execute_fund_community_pool` functions in `src/contract.rs:110` and `158` the `info.funds` sent with the message have minimal processing to ensure that both a `timeout_fee` and `ack_fee` are sent, but there is no validation to ensure that their amounts exceed the configured minimum refund fees of the minimum required fees of the `FeeRefunder` module of the Neutron chain. We classify this issue as informational because this error will ultimately be handled further in the execution of the messages on the Neutron chain, but it does impact user experience by potentially providing less specific error messages.

Recommendation

We recommend enforcing minimum fee values at the contract level to ensure that the fee amounts exceed the minimum required fees of the FeeRefunder module.

4. ICA message handling can be improved

RISK IMPACT: INFO	STATUS: ACKNOWLEDGED

Description

The creation of `ica_msg` in `src/contract.rs:175` can be simplified and be made more readable. Since `MsgFundCommunityPool` implements `prost::Message`, the value of `any_msg` in `line 195` can be directly added with `ica_msg.encode_to_vec().into()` rather than the current method of creating a byte buffer in `lines 182-191`

Recommendation

In `line 195`, set `ProtobufAny.value` to `ica_msg.encode_to_vec().into()` which will allow for the removal of `lines 182-191`.

5. Contract does not ensure `ibc_timeout_seconds` is sufficiently large

RISK IMPACT: INFO	STATUS: ACKNOWLEDGED

Description

In the `instantiate` function in `src/contract.rs:57` `ibc_timeout_seconds` should be validated that it is above a configured timeout minimum. If the timeout is set to be too short, it will potentially cause excessive timeouts that reset the IBC channel.

Recommendation

We recommend imposing a minimum `ibc_timeout_seconds` in the `instantiate` function.

6. `ibc_fee_from_funds` ignores excess denoms

RISK IMPACT: INFO	STATUS: ACKNOWLEDGED

Description

The `ibc_fee_from_funds` function in `src/contract.rs:357` will silently ignore excess denoms sent that are not the `NEUTRON_DENOM`. In this case they will not be refunded to the user and will remain in the contract balance.

Recommendation

We recommend validating that only the `NEUTRON_DENOM` is received and to error if excess funds are sent.

7. Avoid emitting default attributes

RISK IMPACT: INFO	STATUS: ACKNOWLEDGED

Description

Most of the execute messages in the scope of this audit implement the default response attributes. This ultimately provides a poor user experience and also makes the contract events difficult to index for block explorers and other tools that rely on attributes and events being emitted.

Recommendation

We recommend adding descriptive attributes to the execute message that both emit the action and the state changes that occurred during the message execution.

Document Control

Version	Date	Notes
-	2nd November 2023	Security audit commencement date.
0.1	12th November 2023	Initial report with identified findings delivered.
0.5	14th November 2023	Fixes remediations implemented and reviewed.
1.0	18th November 2023	Audit completed, final report delivered.

Appendices

A. Appendix – Risk assessment methodology

SCV-Security employs a risk assessment methodology to evaluate vulnerabilities and identified issues. This approach involves the analysis of both the LIKELIHOOD of a security incident occurring and the potential IMPACT if such an incident were to happen. For each vulnerability, SCV-Security calculates a risk level on a scale of 5 to 1, where 5 denotes the highest likelihood or impact. Consequently, an overall risk level is derived from combining these two factors, resulting in a value from 10 to 1, with 10 signifying the most elevated level of security risk

Risk Level	Range
CRITICAL	10
SEVERE	From 9 to 8
MODERATE	From 7 to 6
LOW	From 5 to 4
INFORMATIONAL	From 3 to 1

LIKELIHOOD and **IMPACT** would be individually assessed based on the below:

Rate	LIKELIHOOD	IMPACT
5	**Extremely Likely**	Could result in severe and irreparable consequences.
4	**Likely**	May lead to substantial impact or loss.
3	**Possible**	Could cause partial impact or loss on a wide scale.
2	**Unlikely**	Might cause temporary disruptions or losses.
1	**Rare**	Could have minimal or negligible impact.

B. Appendix – Report Disclaimer

This report should not be regarded as an "endorsement" or "disapproval" of any specific project or team. These reports do not indicate the economics or value of any "product" or "asset" created by a team or project that engages SCV-Security for a security review. The audit report does not make any statements or warranties about the code's utility, safety, suitability of the business model, regulatory compliance of the business model, or any other claims regarding the fitness of the implementation for its purpose or its bug-free status. The audit documentation is intended 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 damages arising from or in connection with this audit report. Copyright of this report remains with SCV-Security.

THANK YOU FOR CHOOSING

SCV SECURITY

scv.services

contact@scv.services