AUDIT REPORT

Ark Protocol

ICS-721

Prepared by SCV-Security

On 8th February 2024

Table of Contents

Introduction

SCV has been engaged by Ark Protocol 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

Ark Protocol implements a new specification (ICS-721) written in CosmWasm that permits Non-fungible token (NFTs) to be moved across blockchains using Cosmos IBC and compatible blockchains. It extends the cw-721 standard by incorporating both incoming and outgoing proxies.

Submitted Codebase

ics721	
Repository	https://github.com/public-awesome/cw-ics721
Commit	`f1dfefc71c3ace567a5b79e98100ee17d9cfcc5d`
Branch	`main`
ics721-plus	
Repository	https://github.com/arkprotocol/ics721-plus
Commit	`af453532010ec80a25e7e3bfbe9e29f09dcd7a15`
Branch	`main`
ics721-proxy	
Repository	https://github.com/arkprotocol/cw-ics721-proxy
Commit	`58ac2ad8dcf70751975758d8b8925f5b009ddaa2`
Branch	`main`

Revision Codebase

ics721	
Repository	https://github.com/public-awesome/cw-ics721
Commit	`ad8f7b470b141d59cbab6ff8bfc4d3c01da3ac5f`
Branch	`v0.1.8`
ics721-plus	
Repository	https://github.com/arkprotocol/ics721-plus
Commit	`8a0ccbca1386231ef2abbfdc9b04d4e422480e09`
Branch	`v0.1.4`
ics721-proxy	
Repository	https://github.com/arkprotocol/cw-ics721-proxy
Commit	`9c2502d10d0c3ca15c92616b0346b1389912b458`
Branch	`v0.1.1`

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 Ark Protocol. 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 organisation.
- **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 categorised 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**	Ark protocol provided sufficient readme documentation, and blog posts as well as code comments that assisted in understanding the codebase
Coverage	**SUFFICIENT**	Testing coverage is considered sufficient, although there is room for improvement as the current coverage: • cw-ics721 54.69% coverage • cw-ics721-proxy 64.58% coverage
Readability	**SUFFICIENT**	The codebase had good readability overall and utilised many Rust and CosmWasm best practices.
Complexity	**SUFFICIENT**	The audit had a moderate level of complexity due to its IBC functionality and that its functional code spanned multiple repositories.

Findings Summary

Summary Title	Risk Impact	Status
Ics721OutgoingProxy rate limits can be bypassed	LOW	ACKNOWLEDGED
Update misleading config updates	INFO	ACKNOWLEDGED
It is not best practice to overwrite `info.sender`	INFO	RESOLVED
Remove key-value duplication in maps	INFO	RESOLVED
Define ambiguous memo field for `NonFungibleTokenPacketData`	INFO	RESOLVED
Spelling error found in codebase	INFO	RESOLVED

Audit Observations

The audit observations section is intended to present potential findings that are related to the underlying design of the protocol and would require underlying design changes to remediate that may change the overall functioning of the protocol. SCV asks that the client formulate responses to add context to validate or invalidate the following concerns.

1. Metadata is not preserved when NFTs are transferred across chains

Currently, when NFTs are transferred from one blockchain to another using the ICS721 contract, the metadata of individual NFTs is not transferred. Only generic data about the collection is transferred, not the specific data contained in the extension of each `token_id`. This is due to different NFT collections possibly implementing custom extensions, making the type indeterminate. However, it is feasible to deserialize the extension into the `serde_cw_value::Value` type using the `cw721::msg::QueryMsg::NftInfo` query. This information can then be included in the `NonFungibleTokenPacketData` for each `token_id`. Subsequently, when the packet is received on the destination blockchain's contract and the voucher is minted, a cw721 with the `serde_cw_value::Value` extension can be utilized instead of a `cw721-base` (which has an extension as `Empty`). This enables the minting of the voucher with the specific extension for that token_id, preserving the same on-chain information as the original NFT. Moreover, this method is highly advantageous for potential integrations of these vouchers in other projects, particularly on the frontend side, which would not require interfacing with other contracts or blockchains to ascertain the traits or specific information related to that `token_id`.

Notes

The Ark team notes that onchain metadata has been intentionally not been considered in version one for ICS721 and it is planned for next version at https://github.com/public-awesome/cw-ics721/issues/85.

2. Additional vulnerabilities identified during the audit engagement

During the audit engagement, Ark team identified vulnerabilities that could negatively impact the implementation and its functionality. The root cause of the vulnerability relates to an inefficient SubMessage not triggered in case of an error or failure, that would result in a NFT being minted and consequently locked in the contract rather than be transferred to its destination or burned. The Ark team performed a code refactored addressing the underlying root cause and simplified the messages struct, improving the code readability and composability. The PR containing changes is available here.

SCV-Security reviewed and concluded that the identified vulnerabilities were not exploitable, however, remediations were effectively applied, reviewed and tested to ensure the contract functionality remains correct and secure.

Findings Technical Details

1. `Ics721OutgoingProxy` rate limits can be bypassed

RISK IMPACT: LOW	STATUS: ACKNOWLEDGED

Revision Notes

The client has acknowledged this finding and states that rate limits are not currently intended to be enforced and they serve as a proof of concept in the current version of the codebase. In case of need, a feature request has been created here: https://github.com/arkprotocol/ics721-plus/issues/8.

Description

The `Ics721OutgoingProxy::assert_rate_limit` method in `ics721-plus/packages/cw-ics721-outgoing-proxy-whitelist/src/lib.rs:500` asserts usage limits for `CW721-ICS` per user wallet. However, this limit can be bypassed by users transferring the NFT to a new wallet before further usage. This allows exceeding the defined limits. While the rate limit provides some friction, a motivated user can circumvent it through wallet transfers. This reduces the effectiveness of the imposed limits.

Recommendation

We recommend clearly defining the intentions of the rate-limiting mechanism and potentially considering the alternative such as rate-limiting the NFT itself rather than the sender. This would allow for more concrete rate limiting.

2. Update misleading config updates

RISK IMPACT: INFO	STATUS: ACKNOWLEDGED

Revision Notes

The Ark team has created a GitHub issue at #9 that follows SCV's recommendation to rename them to a "response structs" which effectively eliminates the confusion of not being config states.

Description

The `initialize` and `update_config` functions in `cw-ics721:packages/cw-ics721-outgoing-proxy-whitelist/src/lib.rs:227` and `286` perform operations on a mutable `config`. But the config is never saved. While this does not present a security concern to Ark protocol, these changes to a mutable `config` could present future maintainability issues for the protocol.

In CosmWasm contracts `config` is generally a reserved term to represent the contract's `CONFIG` state. It appears that these updates are made with the sole purpose of emitting state changes within the attributes.

Recommendation

We recommend recording the changes to the `Ics721OutgoingProxy` in a more clearer manner by removing or justifying the usage of the multiple `config` variable.

3. It is not best practice to overwrite `info.sender`

RISK IMPACT: INFO	STATUS: RESOLVED

Description

The code at `cw-ics721:packages/cw-ics721-outgoing-proxy-whitelist/src/lib.rs:442` and `cw-ics721:packages/ics721/src/execute.rs:127` overwrites the `info.sender` field when processing messages. This can result in the loss of critical metadata about the original sender of the message. Additionally, it can impact the readability and maintainability of the codebase.

Overwriting `info.sender` goes against best practices for preserving metadata and can make debugging and auditing more difficult. The original sender information should be preserved and new sender info appended or stored separately.

Recommendation

We recommend avoiding directly overwriting `info.sender` fields when processing messages. The intended value can be represented in another variable. If it is absolutely necessary to overwrite `MessageInfo` it should be explicitly defined in the code comments wherever it occurs.

We recommend changing ContractInstantiateInfo::into_wasm_msg to return (WasmMsg, Addr), where WasmMsg is WasmMsg::Instantiate2 and Addr is the address of the contracts instantiated and cast it into CosmoMsg instead of SubMsg.

4. Remove key-value duplication in maps

RISK IMPACT: INFO	STATUS: RESOLVED

Description

In the `cw-ics721` package, specifically in state.rs at `lines 22` and `25`, there are two `cw-storage-plus::Maps` named `CLASS_ID_TO_NFT_CONTRACT` and `NFT_CONTRACT_TO_CLASS_ID`. These maps are responsible for tracking the `class_id` and `collection_addr` in a reciprocal manner. However, this setup leads to redundant operations since any modification to these values necessitates updates in both maps.

Recommendation

A more efficient approach would be to use `cw-storage-plus::IndexMap`, where `address` is the primary key and `class_id` is uniquely indexed using `cw-storage-plus::UniqueIndex`, wrapping the class_id into a struct, and implementing cw-storage-plus::IndexList for an indexer tracker struct, reducing redundancy and improving code readability

5. Spelling error found in codebase

RISK IMPACT: INFO	STATUS: RESOLVED

Description

The code at `contracts/cw-ics721-incoming-proxy-base/src/contract.rs:37` emits the key attribute "`orgin`" when it should be "`origin`" instead.

Recommendation

We recommend correcting this spelling error.

Document Control

Version	Date	Notes
-	8th January 2024	Security audit commencement date.
0.1	16th January 2024	Initial report with identified findings delivered.
0.5	20th January 2024 to 3rd February 2024	Fixes remediations implemented and reviewed.
1.0	8th February 2024	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