Axelar Interchain Token Service (ITS) Security Assessment

Axelar

Version 1.0 – November 25, 2024

Prepared By
Kevin Henry
Parnian Alimi

Prepared For
Christian Gorenflo
Milap Sheth

1 Executive Summary

Synopsis

During the fall of 2024, Axelar engaged NCC Group to conduct a security assessment of the Interchain Token Service (ITS) implementation. ITS is designed to bridge ERC20 tokens between multiple chains. This engagement was comprised of 2 components:

- Axelarnet Gateway and Interchain Token Service (AKA ITS hub) CosmWasm smart contracts,
- Changes to the Axelar core to support ITS.

The review was delivered by 2 consultants with a total effort of 5 person-days.

Scope

The Axelar core repository and the Amplifier protocol and implementation have been previously reviewed by NCC Group. This engagement focused on the added support for ITS.

The review of Axelar core changes since version 1.0.0 included:

- Review of the unique message id generation for the transactions that the Axelarnet Gateway contract handles. Including the changes to:
  - `app/keepers.go`
  - `x/nexus/keeper/general_message.go`
  - `x/nexus/keeper/msg_id_generator.go`
  - `x/nexus/keeper/wasm_querier.go`
  - `x/nexus/keeper/wasmer_engine.go`
- Review of the relevant refactoring changes to the `axelarnet` and the `nexus` modules. Including the changes to:
  - `x/axelarnet/keeper/message_route.go`
  - `x/axelarnet/keeper/migrate.go`
  - `x/axelarnet/keeper/msg_server.go`
  - `x/axelarnet/keeper/message_handler.go`
  - `x/axelarnet/keeper/module.go`
  - `x/nexus/exported/types.go`
  - `x/nexus/keeper/lockable_asset.go`
  - `x/nexus/types/types.go`

On the Amplifier side, the `interchain-token-service` and the `axelarnet-gateway` CosmWasm contracts, on commit `ada9e92` were reviewed in their entirety.

NCC Group used the ITS Token Hub document to guide the review and assess various message flow scenarios.

Key Findings

One low severity finding was identified which describes a discrepancy between the implementation and a code comment which could, if overlooked, hinder future development. In addition, one informational finding regarding the Amplifier repository's dependency management is included for reference.

2 Table of Findings

For each finding, NCC Group uses a composite risk score that takes into account the severity of the risk, application's exposure and user population, technical difficulty of exploitation, and other factors.

Title	Status	ID	Risk
Potentially Unsafe Counter Management During Message ID Generation	New	G7Q	Low
Vulnerable and Outdated Dependencies	New	UL4	Info

3 Finding Details

Low

Potentially Unsafe Counter Management During Message ID Generation

Overall Risk	Low	**Finding ID**	NCC-E010021-G7Q
Impact	Medium	**Component**	axelar-core
Exploitability	None	**Category**	Other
		Status	New

Impact

Failure to generate unique nonces may lead to reuse, thereby compromising security assumptions and potentially enabling replay or re-entry attacks.

Description

As part of the Interchain Token Service, changes were made that ensure that generated message IDs are unique, serving as a safeguard against re-entry attacks and nested message calls. A small set of counter-related utility functions are introduced, and a wrapper ensures that the nonce counter is incremented on every query.

The following 3 utility functions are used to fetch or increment the message ID as needed:

```
18 // IncrID increments the nonce
19 func (k Keeper) IncrID(ctx sdk.Context) {
20   utils.NewCounter[uint64](messageNonceKey, k.getStore(ctx)).Incr(ctx)
21 }
22
23 // nextID returns the transaction hash of the current transaction and the incremented nonce
24 func (k Keeper) nextID(ctx sdk.Context) ([32]byte, uint64) {
25   return getTxHash(ctx), utils.NewCounter[uint64](messageNonceKey,
     ↳ k.getStore(ctx)).Incr(ctx)
26 }
27
28 // CurrID returns the current transaction hash and index
29 func (k Keeper) CurrID(ctx sdk.Context) ([32]byte, uint64) {
30   return getTxHash(ctx), utils.NewCounter[uint64](messageNonceKey,
     ↳ k.getStore(ctx)).Curr(ctx)
31 }
```

Figure 1: axelar-core/x/nexus/keeper/msg_id_generator.go

As highlighted, the `nextID()` function returns the next message ID by calling `Incr()` on the underlying counter. However, the `Incr()` function is implemented as a *pre-increment*, where the current value of the counter is returned *before* being incremented.

```
0 // Incr increments the counter and returns the value before the increment
1 func (c Counter[T]) Incr(ctx sdk.Context) T {
2   curr := c.Curr(ctx)
3   c.store.SetRawNew(c.key, convert.IntToBytes(curr+1))
4
5   return curr
6 }
```

Figure 2: axelar-core/utils/counter.go

Therefore, the implemented behavior of `Incr()` and `nextID()` contradicts the described behavior in the comments.

In the reviewed code, the existing usages appear to be safe. Each query will be associated with a call to `IncrID()`, thereby incrementing the counter. The incorrectly documented `nextID()` is only called from `GenerateMessageID()`, which, in the current implementation will return an ID that has not been used previously.

The risk with the above approach is that future functionality or usage of the functions may not use them safely. For example, it may not be obvious that `CurrID()` and `nextID()` will return the same value, which could lead to incorrect assumptions or usage in the future.

Recommendation

- Ensure that the implemented behavior is intended, i.e., that `nextID()` should return the current value of the counter rather than the incremented value of the counter.
- Update the comment on `nextID()` to correctly describe the implemented behavior.

Location

axelar-core/x/nexus/keeper/msg_id_generator.go

Info

Vulnerable and Outdated Dependencies

Overall Risk	Informational	**Finding ID**	NCC-E010021-UL4
Impact	Low	**Component**	axelar-amplifier
Exploitability	None	**Category**	Patching
		Status	New

Impact

Attackers may use public security advisories to identify and exploit vulnerabilities within the application. Even if vulnerabilities are not exploitable, the presence of outdated or vulnerable dependencies may affect the reputation and the perceived security posture of the application.

Description

During previous engagements, similar findings were presented to document vulnerable and outdated dependencies. Axelar has configured a GitHub Action to forward security advisories to a Slack channel for monitoring on a weekly basis; see *dependabot-vulns-to-slack.yaml*. For completeness, the current cargo audit results are summarized in this finding.

The following crates result in `cargo audit` **vulnerabilities**:

- `curve25519-dalek 3.2.0`
- `rsa 0.8.2`
- `rustls 0.19.1`, `0.21.7`
- `webpki 0.21.4`
- `zerovec 0.10.2`
- `zerovec-derive 0.10.2`

The following crates result in `cargo audit` *warnings*:

- `derivative 2.2.0`
- `difference 2.0.0`
- `dirs 5.0.1`
- `instant 0.1.3`
- `proc-macro-error 1.0.4`
- `yaml-rust 0.4.5`
- `bytemuck 1.16.0`
- `bytes 1.6.0`
- `futures-util 0.3.30`
- `move-bytecode-verifier 0.1.0`
- `move-command-line-common 0.1.0`
- `move-coverage 0.1.0`
- `move-ir-to-bytecode 0.1.0`
- `move-symbol-pool 0.1.0`
- `zerovec 0.10.2`
- `zerovec-derive 0.10.2`

All evidence suggests that the cargo audit notifications are monitored and evaluated by the Axelar team, as the number of vulnerable crates has decreased since the previous review. Therefore, this finding is considered informational, with the recommendation to continue actively monitoring Rust security advisories.

Recommendation

- Ensure that vulnerable dependency notifications are working and that appropriate processes are in place to respond to them.
- Consider using a tool such as `cargo deny` to enforce an allow list of vulnerable dependencies. This would enforce a proactive action to continue in the presence of a vulnerable crate, rather than a reactive action to respond to a vulnerable crate.

Location

dependabot-vulns-to-slack.yaml

4 Finding Field Definitions

The following sections describe the risk rating and category assigned to issues NCC Group identified.

Risk Scale

NCC Group uses a composite risk score that takes into account the severity of the risk, application's exposure and user population, technical difficulty of exploitation, and other factors. The risk rating is NCC Group's recommended prioritization for addressing findings. Every organization has a different risk sensitivity, so to some extent these recommendations are more relative than absolute guidelines.

Overall Risk

Overall risk reflects NCC Group's estimation of the risk that a finding poses to the target system or systems. It takes into account the impact of the finding, the difficulty of exploitation, and any other relevant factors.

Rating	Description
Critical	Implies an immediate, easily accessible threat of total compromise.
High	Implies an immediate threat of system compromise, or an easily accessible threat of large-scale breach.
Medium	A difficult to exploit threat of large-scale breach, or easy compromise of a small portion of the application.
Low	Implies a relatively minor threat to the application.
Informational	No immediate threat to the application. May provide suggestions for application improvement, functional issues with the application, or conditions that could later lead to an exploitable finding.

Impact

Impact reflects the effects that successful exploitation has upon the target system or systems. It takes into account potential losses of confidentiality, integrity and availability, as well as potential reputational losses.

Rating	Description
High	Attackers can read or modify all data in a system, execute arbitrary code on the system, or escalate their privileges to superuser level.
Medium	Attackers can read or modify some unauthorized data on a system, deny access to that system, or gain significant internal technical information.
Low	Attackers can gain small amounts of unauthorized information or slightly degrade system performance. May have a negative public perception of security.

Exploitability

Exploitability reflects the ease with which attackers may exploit a finding. It takes into account the level of access required, availability of exploitation information, requirements relating to social engineering, race conditions, brute forcing, etc, and other impediments to exploitation.

Rating	Description
High	Attackers can unilaterally exploit the finding without special permissions or significant roadblocks.

Rating	Description
Medium	Attackers would need to leverage a third party, gain non-public information, exploit a race condition, already have privileged access, or otherwise overcome moderate hurdles in order to exploit the finding.
Low	Exploitation requires implausible social engineering, a difficult race condition, guessing difficult-to-guess data, or is otherwise unlikely.

Category

NCC Group categorizes findings based on the security area to which those findings belong. This can help organizations identify gaps in secure development, deployment, patching, etc.

Category Name	Description
Access Controls	Related to authorization of users, and assessment of rights.
Auditing and Logging	Related to auditing of actions, or logging of problems.
Authentication	Related to the identification of users.
Configuration	Related to security configurations of servers, devices, or software.
Cryptography	Related to mathematical protections for data.
Data Exposure	Related to unintended exposure of sensitive information.
Data Validation	Related to improper reliance on the structure or values of data.
Denial of Service	Related to causing system failure.
Error Reporting	Related to the reporting of error conditions in a secure fashion.
Patching	Related to keeping software up to date.
Session Management	Related to the identification of authenticated users.
Timing	Related to race conditions, locking, or order of operations.

5 Contact Info

The team from NCC Group has the following primary members:

- Parnian Alimi – Consultant
  parnian.alimi@nccgroup.com
- Kevin Henry – Consultant
  kevin.henry@nccgroup.com
- Javed Samuel – Practice Director, Cryptography Services
  javed.samuel@nccgroup.com

The team from Axelar has the following primary members:

- Christian Gorenflo
  christian@interoplabs.io
- Milap Sheth
  milap@interoplabs.io
- Liana Spano
  liana@interoplabs.io