Security Audit Report

Axelar solana-operators (2)

Solana program

by ARDA

on April 15, 2026

Table of Contents

Disclaimer **3**

Terminology **4**

Objective **4**

Summary **5**

Code Issues & Recommendations **6**

C1: Redundant operator field stored in OperatorAccount 6

C2: Superfluous constraint check in remove_operator 7

Disclaimer

The report makes no statements or warranties, either expressed or implied, regarding the security of the code, the information herein or its usage. It also cannot be considered as a sufficient assessment regarding the utility, safety and bugfree status of the code, or any other statements.

This report does not constitute legal or investment advice. It is for informational purposes only and is provided on an "as-is" basis. You acknowledge that any use of this report and the information contained herein is at your own risk. The authors of this report shall not be liable to you or any third parties for any acts or omissions undertaken by you or any third parties based on the information contained herein.

Terminology

Code: The code with which users interact.

Inherent risk: A risk for users that comes from a behavior inherent to the code's design.

Inherent risks only represent the risks inherent to the code's design, which are a subset of all the possible risks. **No inherent risk doesn't mean no risk.** It only means that no risk inherent to the code's design has been identified. Other kind of risks could still be present. For example, the issues not fixed incur risks for the users, or the upgradability of the code might also incur risks for the users.

Issue: A behavior unexpected by the users or by the project, or a practice that increases the chances of unexpected behaviors to appear.

Critical issue: An issue intolerable for the users or the project, that must be addressed.

Major issue: An issue undesirable for the users or the project, that we strongly recommend to address.

Medium issue: An issue uncomfortable for the users or the project, that we recommend to address.

Minor issue: An issue imperceptible for the users or the project, that we advise to address for the overall project security.

Objective

Our objective is to share everything we have found that would help assessing and improving the safety of the code:

1. The **inherent risks** of the code, labelled R1, R2, etc.
2. The **issues** in the **code**, labelled C1, C2, etc.
3. The **issues** in the **testing** of the code, labelled T1, T2, etc.
4. The **issues** in the **other** parts related to the code, labelled O1, O2, etc.
5. The **recommendations** to address each issue.

Summary

Blockchain: Solana

Initial scope

- **Repository:** https://github.com/axelarnetwork/axelar-amplifier-solana
- **Commit:** `30e4195a839ded2d7a695b2e9e71e06242a43b58`
- **Path(s):** `./programs/solana-axelar-operators`

Final scope

- **Repository:** https://github.com/axelarnetwork/axelar-amplifier-solana
- **Commit:** `3e7a6f81ea41bc1eafb78dd10acbdbb589bd1a00`
- **Path(s):** `./programs/solana-axelar-operators`

0 inherent risk in the final scope

0 issue in the final scope

2 issues reported in the initial scope and 0 remaining in the final scope.

Severity	Reported			Remaining		
	Code	Test	Other	Code	Test	Other
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	0	0	0	0	0	0
Minor	2	0	0	0	0	0

Code Issues & Recommendations

C1: Redundant operator field stored in OperatorAccount

Severity: Minor **Status:** Fixed

Location

```
src/instructions/add_operator.rs
    add_operator
```

Description

The `OperatorAccount` PDA stores the operator's public key in its `operator` field, but this is redundant because the operator key is already part of the PDA seeds:

```
[OperatorAccount::SEED_PREFIX, operator.key()]
```

Therefore, the mere existence of the `OperatorAccount` PDA guarantees the operator's identity.

The stored field is only used in `remove_operator` via a constraint check, but this check is superfluous (see C2: Superfluous constraint check in remove_operator).

Additionally, in the Gas Service, the `operator` field is not even checked when validating operator accounts, further confirming its redundancy.

Recommendation

We recommend removing the `operator` field from `OperatorAccount` and the corresponding constraint in `remove_operator`, since the PDA seeds already encode this information.

C2: Superfluous constraint check in remove_operator

Severity: Minor **Status:** Fixed

Location

`src/instructions/remove_operator.rs`
`RemoveOperator`

Description

In the `RemoveOperator` account struct, the `operator_account` has an Anchor constraint checking `operator_account.operator == operator_to_remove.key()`. This check is superfluous because the PDA seeds already include `operator_to_remove.key()`:

```
#[account(
    mut,
    close = owner,
    seeds = [
        OperatorAccount::SEED_PREFIX,
        operator_to_remove.key().as_ref(),
    ],
    bump = operator_account.bump,
    // Superfluous: already guaranteed by seeds
    constraint = operator_account.operator ==
operator_to_remove.key()
        @ ErrorCode::InvalidOperator
)]
```

Indeed, the PDA derivation guarantees that the `operator_account` corresponds to `operator_to_remove`, making the explicit constraint redundant.

Recommendation

We recommend removing the superfluous constraint check, as the PDA seed derivation already enforces this invariant.