Security Assessment Report

Express Relay Solana

Incremental Review

January 17, 2025

Summary

The Sec3 team (formerly Soteria) was engaged to conduct an incremental review of the Pyth Express Relay (Solana) smart contract.

The artifact of the audit was the source code of the following programs, excluding tests, in https://github.com/pyth-network/per/tree/f3026e6/contracts/svm.

The initial audit focused on the following versions and revealed 1 question.

program	type	commit
express-relay	Solana	f3026e6 → 76d65c2

This report provides a detailed description of the findings and their respective resolutions.

Table of Contents

Result Overview

Issue	Impact	Status
EXPRESS-RELAY		
[Q-01] Inconsistent "split_relayer" usage	Question	Resolved

Findings in Detail

EXPRESS-RELAY
[Q-01] Inconsistent "split_relayer" usage

In the "submit_bid" instruction, the relayer fee is calculated as "(bid_amount - fee_router) * split_relayer":

```
/* contracts/svm/programs/express_relay/src/utils.rs */
148 | let fee_relayer = bid_amount
149 |     .saturating_sub(fee_router)
150 |     .checked_mul(split_relayer)
151 |     .ok_or(ProgramError::ArithmeticOverflow)?
152 |     / FEE_SPLIT_PRECISION;
```

However, the "swap" instruction calculates the relayer fee using "amount * swap_platform_fee_bps * split_relayer":

```
/* contracts/svm/programs/express_relay/src/swap.rs */
126 | let platform_fee = amount
127 |     .checked_mul(self.swap_platform_fee_bps)
128 |     .ok_or(ProgramError::ArithmeticOverflow)?
129 |     / FEE_SPLIT_PRECISION;
130 | let relayer_fee = platform_fee
131 |     .checked_mul(self.split_relayer)
132 |     .ok_or(ProgramError::ArithmeticOverflow)?
133 |     / FEE_SPLIT_PRECISION;
```

The same "split_relayer" variable is used differently in the two instructions. Is this the intended behavior?

Resolution

The team confirmed that this is intended.

In "submit_bid", there is an implicit platform fee equal to "(bid_amount - fee_router)". This fee is then split between "fee_receiver_relayer" and "express_relay_metadata".

In "swap", the platform fee is explicit and calculated as "amount * swap_platform_fee_bps".

The platform fee distribution between `"fee_receiver_relayer"` and `"express_relay_metadata"` should remain consistent across both operations.

Appendix: Methodology and Scope of Work

Assisted by the Sec3 Scanner developed in-house, the manual audit particularly focused on the following work items:

- Check common security issues.
- Check program logic implementation against available design specifications.
- Check poor coding practices and unsafe behavior.
- The soundness of the economics design and algorithm is out of scope of this work

DISCLAIMER

The instance report ("Report") was prepared pursuant to an agreement between Coderrect Inc. d/b/a Sec3 (the "Company") and Pyth Data Association dba Pyth Network (the "Client"). This Report solely includes the results of a technical assessment of a specific build and/or version of the Client's code specified in the Report ("Assessed Code") by the Company. The sole purpose of the Report is to provide the Client with the results of the technical assessment of the Assessed Code. The Report does not apply to any other version and/or build of the Assessed Code. Regardless of the contents of the Report, the Report does not (and should not be interpreted to) provide any warranty, representation or covenant that the Assessed Code: (i) is error and/or bug free, (ii) has no security vulnerabilities, and/or (iii) does not infringe any third-party rights. Moreover, the Report is not, and should not be considered, an endorsement by the Company of the Assessed Code and/or of the Client. Finally, the Report should not be considered investment advice or a recommendation to invest in the Assessed Code and/or the Client.

This Report is considered null and void if the Report (or any portion thereof) is altered in any manner.

ABOUT

The Sec3 audit team comprises a group of computer science professors, researchers, and industry veterans with extensive experience in smart contract security, program analysis, testing, and formal verification. We are also building automated security tools that incorporate static analysis, penetration testing, and formal verification.

At Sec3, we identify and eliminate security vulnerabilities through the most rigorous process and aided by the most advanced analysis tools.

For more information, check out our website and follow us on twitter.