Security Assessment Report

Express Relay (Solana)

October 07, 2024

Summary

The Sec3 team (formerly Soteria) was engaged to conduct a thorough security analysis of the Express Relay (Solana) smart contracts.

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

The initial audit focused on the following versions and revealed 1 issues or questions.

program	type	commit
express-relay	Solana	`ffbdff2cbb9ffecccff658bfa66f3cc1f9364560`

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

Table of Contents

Result Overview

Issue	Impact	Status
EXPRESS-RELAY		
[I-01] Consider moving "permission" into instruction data	Info	Acknowledged

Findings in Detail

EXPRESS-RELAY

[I-01] Consider moving "permission" into instruction data

The "permission" in the account list does not correspond to any key pair or account. The program will not attempt to verify if it is a signer, nor will it try to read it as an account. Based on the auction server's code, it looks like the permission can be any 32-byte data, with no checks or restrictions in place.

```
/* contracts/svm/programs/express_relay/src/lib.rs */
243 | pub struct SubmitBid<'info> {
249 |     /// CHECK: this is the permission_key
250 |     pub permission: UncheckedAccount<'info>,
272 | }
274 | #[derive(Accounts)]
275 | pub struct CheckPermission<'info> {
280 |     /// CHECK: this is the permission_key
281 |     pub permission: UncheckedAccount<'info>,
285 | }
```

Therefore, the "permission" could be removed from the account list. It would be better to pass it through the instruction data.

If multiple transactions reference the same account with at least one write, they cannot be processed in parallel. So, introducing unnecessary accounts into the account list could reduce the protocol's TPS, depending on how frequently the instruction that writes to the "permission" account appears.

Resolution

The team acknowledged this finding and clarified that the "permission" account is never intended to be written to, so there won't be any contestations.

Additionally, including it as an account rather than as instruction data saves transaction space, as repeated accounts are cached in the static accounts list and incur only a one-byte overhead.

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.