Security Assessment Report

Monaco Protocol v0.6.0

February 27, 2023

Summary

The sec3 team (formerly Soteria) was engaged to do a thorough security analysis of the Monaco Protocol Solana smart contract in https://github.com/MonacoProtocol/protocol. The initial audit was done on the source code of the following version

- **Contract "monaco_protocol":**
  - v0.6.0-rc1, commit 682acc39ba21d0669caa41afe1b0ad024dfc6039

The audit revealed 5 issues or questions. After the initial review, the team responded with the following commits for the post-audit review, which is to validate if the reported issues have been addressed.

  - V0.6.0-rc2, commit c464322e96a706e50827aff6e917b90c59e3cd53
  - V0.6.0-rc3, commit 1054f8fc8050632398e4a78957baa51915debedb
  - V0.6.0-rc4, commit 1b6e6ca172bb3e471a135c0bb4e420d9cf747cea
  - v0.6.0, commit 4d7ad27ecf801a1933130d7b5d7983e93b02e72c

This report describes the findings and resolutions in detail.

Table of Contents

Methodology and Scope of Work 3

Result Overview 4

Findings in Detail 5

[M-1] verify_prices_precision rounding issue 5

[L-1] price_ladder may contain duplicated price items 6

[I-1] Account closure side effects 7

[I-2] Negative numbers allowed in price_ladder 8

[I-3] Market escrow accounts are not closed 9

Methodology and Scope of Work

The sec3 (formerly Soteria) audit team, which consists of Computer Science professors and industrial researchers with extensive experience in Solana smart contract security, program analysis, testing and formal verification, performed a comprehensive manual code review, software static analysis and penetration testing.

Assisted by the sec3 Scanner developed in-house, the audit team particularly focused on the following work items:

- Check common security issues.
    - Missing ownership checks
    - Missing signer checks
    - Signed invocation of unverified programs
    - Solana account confusions
    - Arithmetic over- or underflows
    - Numerical precision errors
    - Loss of precision in calculation
    - Insufficient SPL-Token account verification
    - Missing rent exemption assertion
    - Casting truncation
    - Did not follow security best practices
    - Outdated dependencies
    - Redundant code
    - Unsafe Rust code
- 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

Result Overview

In total, the audit team found the following issues.

MONACO PROTOCOL v0.6.0

Issue	Impact	Status
[M-1] verify_prices_precision rounding issue	Medium	Fixed
[L-1] price_ladder may contain duplicated price items	Low	Fixed
[I-1] Account closure side effects	Informational	Fixed
[I-2] Negative numbers allowed in price_ladder	Informational	Fixed
[I-3] Market escrow accounts are not closed	Informational	Fixed

Findings in Detail

IMPACT – MEDIUM

[M-1] verify_prices_precision rounding issue

This function checks if a price has at most three decimal digits by validating that the formatted price string "format!("{value:.3}")" is larger than the price string format!("{value}"). However, a number with more digits can bypass the check when rounding occurs. For example,

```
/* monaco_protocol/src/instructions/market/create_market.rs */
078 | fn verify_prices_precision(prices: &[f64]) -> Result<()> {
+   |     let v = prices[0];
+   |     let v_0_str = format!("{v:.3}");
+   |     let v_1_str = format!("{v}");
+   |     print!("{} <= {} = {}\n", v_1_str, v_0_str, v_1_str <= v_0_str);
079 |     require!(
080 |         prices
081 |             .iter()
082 |             .all(|&value| format!("{value}") <= format!("{value:.3}")),
083 |         CoreError::MarketPricePrecisionTooLarge
084 |     );
085 |     Ok(())
086 | }

#[test]
fn test_verify_prices_precision() {
    let ok = verify_prices_precision(&vec![1.1118]);
    assert!(ok.is_err());
}
```

Result

```
running 1 test
1.1118 <= 1.112 = true
thread 'instructions::market::create_market::tests::test_verify_prices_precision'
 panicked at 'assertion failed: ok.is_err()',
programs/monaco_protocol/src/instructions/market/create_market.rs:155:9
```

Resolution

This issue has been fixed in PR #19

IMPACT – LOW

[L-1] price_ladder may contain duplicated price items

When initializing the outcome, the price items in the price_ladder only need to pass the precision check, so there can be duplications.

By contrast, when adding prices to the price_ladder, those items will be de-duplicated.

```
/* monaco_protocol/src/instructions/market/create_market.rs */
052 | pub fn initialize_outcome(
053 |     ctx: Context<InitializeMarketOutcome>,
054 |     title: String,
055 |     price_ladder: Vec<f64>,
056 | ) -> Result<()> {
061 |     verify_prices_precision(&price_ladder)?;
068 |     ctx.accounts.outcome.price_ladder = price_ladder;

078 | fn verify_prices_precision(prices: &[f64]) -> Result<()> {
079 |     require!(
080 |         prices
081 |             .iter()
082 |             .all(|&value| format!("{value}") <= format!("{value:.3}")),
083 |         CoreError::MarketPricePrecisionTooLarge
084 |     );
085 |     Ok(())
086 | }
```

Resolution

This issue has been fixed in 93074a01.

IMPACT – INFO

[I-1] Account closure side effects

Those newly added account closure instructions look good by themselves. However, an impropriate sequence of these account closure instructions may introduce undesired side effects on the remaining accounts.

In particular, the order status seems almost independent from the market status (except when creating an order). There are no requirements for when a specific account can be closed. For example, before closing the accounts, even if the market status moves to "ReadyToClose," it is possible to cancel not fully matched orders and get refunds. However, at this point, if crank operators directly close the order together with other accounts, end users will not get refunds.

However, because only authorized market operators can change market status and only authorized crank operators can close accounts, the risk is very low.

Resolution

Additional account settlement and closure requirements have been added in PR #21:

1. The balance of market escrow has to be 0 before allowing a market to move to Settled.
2. The order status has to be settled before being closed.

IMPACT – INFO

[I-2] Negative numbers allowed in price_ladder

The price_ladder validation function currently allows negative numbers.

However, it's still safe since the order price is required to be larger than 1 when initializing orders. It would still be a good idea to filter out price_ladder items that are not larger than 1.

Resolution

The range check has been added in PR #22. This issue has been fixed.

IMPACT – INFO

[I-3] Market escrow accounts are not closed

The market escrow token account is created when creating a market account.

However, the escrow account is not closed when closing the market account.

Resolution

This issue has been fixed in PR #24.

DISCLAIMER

The instance report ("Report") was prepared pursuant to an agreement between Coderrect Inc. d/b/a sec3 (the "Company") and BetDEX Labs (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

Founded by leading academics in the field of software security and senior industrial veterans, sec3 (formerly Soteria) is a leading blockchain security company that currently focuses on Solana programs. We are also building sophisticated 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.

sec3™