FYEO

Security Assessment of the Levana Protocol

Levana Foundation

April 2023
Version 1.1

Presented by:
FYEO Inc.
PO Box 147044
Lakewood CO 80214
United States

Security Level
Strictly Confidential

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

Overview

Levana Foundation engaged FYEO Inc. to perform a Security Assessment of the Levana Protocol.

The assessment was conducted remotely by the FYEO Security Team. Testing took place on March 06 - April 13, 2023, and focused on the following objectives:

- To provide the customer with an assessment of their overall security posture and any risks that were discovered within the environment during the engagement.
- To provide a professional opinion on the maturity, adequacy, and efficiency of the security measures that are in place.
- To identify potential issues and include improvement recommendations based on the results of our tests.

This report summarizes the engagement, tests performed, and findings. It also contains detailed descriptions of the discovered vulnerabilities, steps the FYEO Security Team took to identify and validate each issue, as well as any applicable recommendations for remediation.

Key Findings

The following issues have been identified during the testing period. These should be prioritized for remediation to reduce the risk they pose:

- FYEO-LEVANA-01 – Attackers can modify the position of other users via the trigger order flow
- FYEO-LEVANA-02 – Users can remove collateral without impacting notional size
- FYEO-LEVANA-03 – Users funds can be locked when performing operations that accept native tokens
- FYEO-LEVANA-04 – Users incur additional fees when calling removal updates in the CW20 handler

Based on our review process, we conclude that the reviewed code implements the documented functionality.

Scope and Rules of Engagement

The FYEO Review Team performed a Security Assessment of the Levana Protocol. The following table documents the targets in scope for the engagement. No additional systems or resources were in scope for this assessment.

The source code was supplied through a private repository at https://github.com/Levana-Protocol/levana-perps with the commit hash 92bfa9d7265a7aec8bc2ddc5f476c179985f48d5.

A re-review was performed with the commit hash a08a140a26c1ac0f050b773e1086ca684aa4624c.

Files included in the code review

```
levana/
├── contracts/
│   ├── cw20/
│   │   └── src/
│   │       ├── state/
│   │       │   ├── cw20.rs
│   │       │   └── trading_competition.rs
│   │       ├── contract.rs
│   │       ├── lib.rs
│   │       └── state.rs
│   ├── factory/
│   │   └── src/
│   │       ├── state/
│   │       │   ├── all_contracts.rs
│   │       │   ├── auth.rs
│   │       │   ├── label.rs
│   │       │   ├── liquidity_token.rs
│   │       │   ├── market.rs
│   │       │   ├── position_token.rs
│   │       │   ├── reply.rs
│   │       │   └── shutdown.rs
│   │       ├── contract.rs
│   │       ├── lib.rs
│   │       └── state.rs
│   ├── faucet/
│   │   └── src/
│   │       ├── state/
│   │       │   ├── faucet.rs
│   │       │   ├── owner.rs
│   │       │   ├── tokens.rs
│   │       │   └── trading_competition.rs
│   │       ├── contract.rs
│   │       ├── lib.rs
│   │       └── state.rs
│   ├── liquidity_token/
│   │   └── src/
│   │       ├── state/
│   │       │   ├── kind.rs
│   │       │   └── market.rs
│   │       ├── contract.rs
│   │       ├── lib.rs
│   │       └── state.rs
│   ├── market/
```

Files included in the code review

```
│   │       └── src/
│   │           ├── state/
│   │           │   ├── funding/
│   │           │   │   ├── aggregate_capping.rs
│   │           │   │   └── borrow_fees.rs
│   │           │   ├── history/
│   │           │   │   ├── lp.rs
│   │           │   │   ├── mod.rs
│   │           │   │   └── trade.rs
│   │           │   ├── liquidity/
│   │           │   │   ├── cw20.rs
│   │           │   │   └── stats.rs
│   │           │   ├── position/
│   │           │   │   ├── close.rs
│   │           │   │   ├── cw721.rs
│   │           │   │   ├── liquifund.rs
│   │           │   │   ├── open.rs
│   │           │   │   ├── update.rs
│   │           │   │   └── validate.rs
│   │           │   ├── config.rs
│   │           │   ├── crank.rs
│   │           │   ├── data_series.rs
│   │           │   ├── delta_neutrality_fee.rs
│   │           │   ├── fees.rs
│   │           │   ├── funding.rs
│   │           │   ├── liquidity.rs
│   │           │   ├── market.rs
│   │           │   ├── meta.rs
│   │           │   ├── order.rs
│   │           │   ├── position.rs
│   │           │   ├── sanity.rs
│   │           │   ├── shutdown.rs
│   │           │   ├── stale.rs
│   │           │   ├── status.rs
│   │           │   └── token.rs
│   │           ├── constants.rs
│   │           ├── contract.rs
│   │           ├── lib.rs
│   │           ├── prelude.rs
│   │           └── state.rs
│   ├── position_token/
│   │   └── src/
│   │       ├── state/
```

Files included in the code review

```
│   │       │   └── market.rs
│   │       ├── contract.rs
│   │       ├── lib.rs
│   │       └── state.rs
│   └── tracker/
│       └── src/
│           ├── execute.rs
│           ├── lib.rs
│           ├── lifecycle.rs
│           ├── query.rs
│           └── state.rs
├── packages/
│   ├── diagnostics/
│   │   └── src/
│   │       ├── page/
│   │       │   ├── home/
│   │       │   │   ├── app/
│   │       │   │   │   ├── controls/
│   │       │   │   │   │   ├── actions.rs
│   │       │   │   │   │   ├── dom.rs
│   │       │   │   │   │   ├── mod.rs
│   │       │   │   │   │   └── state.rs
│   │       │   │   │   ├── event_view/
│   │       │   │   │   │   ├── actions.rs
│   │       │   │   │   │   ├── dom.rs
│   │       │   │   │   │   ├── mod.rs
│   │       │   │   │   │   └── state.rs
│   │       │   │   │   ├── graph/
│   │       │   │   │   │   ├── actions.rs
│   │       │   │   │   │   ├── camera.rs
│   │       │   │   │   │   ├── dom.rs
│   │       │   │   │   │   ├── icosahedron.rs
│   │       │   │   │   │   ├── mod.rs
│   │       │   │   │   │   └── state.rs
│   │       │   │   │   ├── stats/
│   │       │   │   │   │   ├── actions.rs
│   │       │   │   │   │   ├── mod.rs
│   │       │   │   │   │   └── state.rs
│   │       │   │   │   ├── actions.rs
│   │       │   │   │   ├── dom.rs
│   │       │   │   │   ├── mod.rs
│   │       │   │   │   └── state.rs
│   │       │   │   ├── init/
```

Files included in the code review

```
│   │       │   │   │   ├── actions.rs
│   │       │   │   │   ├── dom.rs
│   │       │   │   │   ├── mod.rs
│   │       │   │   │   └── state.rs
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── not_found/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   └── mod.rs
│   │       ├── primitives/
│   │       │   ├── button/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── checkbox/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── collapsable/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── dropdown/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── image/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── input/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── overlay/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── progress/
│   │       │   │   ├── dom.rs
```

Files included in the code review

```
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── range/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── table/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   ├── tabs/
│   │       │   │   ├── dom.rs
│   │       │   │   ├── mod.rs
│   │       │   │   └── state.rs
│   │       │   └── mod.rs
│   │       ├── runner/
│   │       │   ├── exec.rs
│   │       │   └── mod.rs
│   │       ├── bridge.rs
│   │       ├── config.rs
│   │       ├── lib.rs
│   │       ├── prelude.rs
│   │       └── route.rs
│   ├── fuzz/
│   │   └── fuzz_targets/
│   │       └── market.rs
│   ├── msg/
│   │   ├── examples/
│   │   │   └── generate-schema.rs
│   │   └── src/
│   │       ├── constants/
│   │       │   ├── event_key.rs
│   │       │   └── event_val.rs
│   │       ├── contracts/
│   │       │   ├── cw20/
│   │       │   │   ├── entry.rs
│   │       │   │   └── events.rs
│   │       │   ├── factory/
│   │       │   │   ├── entry.rs
│   │       │   │   └── events.rs
│   │       │   ├── faucet/
│   │       │   │   ├── entry.rs
│   │       │   │   ├── error.rs
```

Files included in the code review

```
│   │       │   │   └── events.rs
│   │       │   ├── liquidity_token/
│   │       │   │   └── entry.rs
│   │       │   ├── market/
│   │       │   │   ├── position/
│   │       │   │   │   ├── closed.rs
│   │       │   │   │   └── collateral_and_usd.rs
│   │       │   │   ├── config.rs
│   │       │   │   ├── crank.rs
│   │       │   │   ├── delta_neutrality_fee.rs
│   │       │   │   ├── entry.rs
│   │       │   │   ├── fees.rs
│   │       │   │   ├── history.rs
│   │       │   │   ├── liquidity.rs
│   │       │   │   ├── order.rs
│   │       │   │   ├── position.rs
│   │       │   │   └── spot_price.rs
│   │       │   ├── position_token/
│   │       │   │   ├── entry.rs
│   │       │   │   └── events.rs
│   │       │   ├── tracker/
│   │       │   │   ├── entry.rs
│   │       │   │   └── events.rs
│   │       │   ├── cw20.rs
│   │       │   ├── factory.rs
│   │       │   ├── faucet.rs
│   │       │   ├── liquidity_token.rs
│   │       │   ├── market.rs
│   │       │   ├── position_token.rs
│   │       │   └── tracker.rs
│   │       ├── bridge.rs
│   │       ├── constants.rs
│   │       ├── contracts.rs
│   │       ├── lib.rs
│   │       ├── prelude.rs
│   │       ├── shutdown.rs
│   │       └── token.rs
│   ├── multi_test/
│   │   ├── src/
│   │   │   ├── arbitrary/
│   │   │   │   ├── funding_payment/
│   │   │   │   │   ├── data.rs
│   │   │   │   │   ├── mod.rs
```

Files included in the code review

```
│   │   │   │   │   ├── runner.rs
│   │   │   │   │   └── strategy.rs
│   │   │   │   ├── lp/
│   │   │   │   │   ├── data.rs
│   │   │   │   │   ├── mod.rs
│   │   │   │   │   ├── runner.rs
│   │   │   │   │   └── strategy.rs
│   │   │   │   ├── position_open/
│   │   │   │   │   ├── data.rs
│   │   │   │   │   ├── mod.rs
│   │   │   │   │   ├── runner.rs
│   │   │   │   │   └── strategy.rs
│   │   │   │   ├── position_update/
│   │   │   │   │   ├── data.rs
│   │   │   │   │   ├── mod.rs
│   │   │   │   │   ├── runner.rs
│   │   │   │   │   └── strategy.rs
│   │   │   │   ├── helpers.rs
│   │   │   │   └── mod.rs
│   │   │   ├── config.rs
│   │   │   ├── cw20_helpers.rs
│   │   │   ├── cw721_helpers.rs
│   │   │   ├── extensions.rs
│   │   │   ├── lib.rs
│   │   │   ├── macros.rs
│   │   │   ├── market_wrapper.rs
│   │   │   ├── market_wrapper_scenarios.rs
│   │   │   ├── position_helpers.rs
│   │   │   ├── response.rs
│   │   │   └── time.rs
│   │   └── tests/
│   │       └── multi_test/
│   │           ├── position/
│   │           │   ├── close.rs
│   │           │   ├── data.rs
│   │           │   ├── infinite_gains.rs
│   │           │   ├── liquidate.rs
│   │           │   ├── max_leverage.rs
│   │           │   ├── minimum_size.rs
│   │           │   ├── misc.rs
│   │           │   ├── mod.rs
│   │           │   ├── open.rs
│   │           │   ├── pending_fees.rs
```

Files included in the code review

```
│   │           │   ├── pnl.rs
│   │           │   ├── take_profit.rs
│   │           │   └── update.rs
│   │           ├── proptest/
│   │           │   ├── funding_payment.rs
│   │           │   ├── lp.rs
│   │           │   ├── mod.rs
│   │           │   ├── position_open.rs
│   │           │   └── position_update.rs
│   │           ├── auth.rs
│   │           ├── crank_fee.rs
│   │           ├── delta_neutrality_fee.rs
│   │           ├── delta_neutrality_ratio.rs
│   │           ├── diagnostic.rs
│   │           ├── fees.rs
│   │           ├── funding.rs
│   │           ├── history.rs
│   │           ├── liquidation_price.rs
│   │           ├── liquidity.rs
│   │           ├── logging.rs
│   │           ├── main.rs
│   │           ├── multi_market.rs
│   │           ├── nft.rs
│   │           ├── order.rs
│   │           ├── sanity.rs
│   │           ├── shutdown.rs
│   │           └── staleness.rs
│   ├── perps-exes/
│   │   └── src/
│   │       ├── bin/
│   │       │   ├── perps-bots/
│   │       │   │   ├── app/
│   │       │   │   │   ├── status_collector/
│   │       │   │   │   │   └── gas_funds.rs
│   │       │   │   │   ├── crank.rs
│   │       │   │   │   ├── factory.rs
│   │       │   │   │   ├── liquidity.rs
│   │       │   │   │   ├── nibb.rs
│   │       │   │   │   ├── price.rs
│   │       │   │   │   ├── status_collector.rs
│   │       │   │   │   ├── trader.rs
│   │       │   │   │   └── utilization.rs
│   │       │   │   ├── endpoints/
```

Files included in the code review

```
│   │       │   │   │   ├── common.rs
│   │       │   │   │   ├── epochs.rs
│   │       │   │   │   ├── factory.rs
│   │       │   │   │   ├── faucet.rs
│   │       │   │   │   ├── markets.rs
│   │       │   │   │   ├── mod.rs
│   │       │   │   │   └── status.rs
│   │       │   │   ├── util/
│   │       │   │   │   ├── markets.rs
│   │       │   │   │   └── mod.rs
│   │       │   │   ├── app.rs
│   │       │   │   ├── cli.rs
│   │       │   │   ├── main.rs
│   │       │   │   └── market_contract.rs
│   │       │   ├── perps-bridge/
│   │       │   │   ├── context.rs
│   │       │   │   ├── handler.rs
│   │       │   │   └── main.rs
│   │       │   ├── perps-deploy/
│   │       │   │   ├── app.rs
│   │       │   │   ├── chain_tests.rs
│   │       │   │   ├── cli.rs
│   │       │   │   ├── factory.rs
│   │       │   │   ├── faucet.rs
│   │       │   │   ├── init_chain.rs
│   │       │   │   ├── instantiate.rs
│   │       │   │   ├── local_deploy.rs
│   │       │   │   ├── localtest.rs
│   │       │   │   ├── main.rs
│   │       │   │   ├── migrate.rs
│   │       │   │   ├── setup_market.rs
│   │       │   │   ├── store_code.rs
│   │       │   │   ├── tracker.rs
│   │       │   │   └── util.rs
│   │       │   └── perps-qa/
│   │       │       ├── cli.rs
│   │       │       ├── discovery.rs
│   │       │       └── main.rs
│   │       ├── types/
│   │       │   ├── mod.rs
│   │       │   └── money.rs
│   │       ├── config.rs
│   │       ├── lib.rs
```

Files included in the code review

```
│   │       └── wallet_manager.rs
│   └── shared/
│       └── src/
│           ├── number/
│           │   ├── convert.rs
│           │   ├── nonzero.rs
│           │   ├── ops.rs
│           │   ├── serialize.rs
│           │   └── types.rs
│           ├── addr.rs
│           ├── auth.rs
│           ├── cosmwasm.rs
│           ├── direction.rs
│           ├── error.rs
│           ├── event.rs
│           ├── leverage.rs
│           ├── lib.rs
│           ├── log.rs
│           ├── market_type.rs
│           ├── max_gains.rs
│           ├── namespace.rs
│           ├── number.rs
│           ├── prelude.rs
│           ├── price.rs
│           ├── response.rs
│           ├── result.rs
│           ├── storage.rs
│           └── time.rs
└── research/
    └── price-manip-modeling/
        └── src/
            ├── amm.rs
            ├── attack.rs
            ├── cli.rs
            ├── config.rs
            ├── main.rs
            ├── perps.rs
            ├── system.rs
            └── types.rs
```

Table 1: Scope

TECHNICAL ANALYSES AND FINDINGS

During the Security Assessment of the Levana Protocol, we discovered:

- 1 finding with CRITICAL severity rating.
- 1 finding with MEDIUM severity rating.
- 2 findings with LOW severity rating.

The following chart displays the findings by severity.

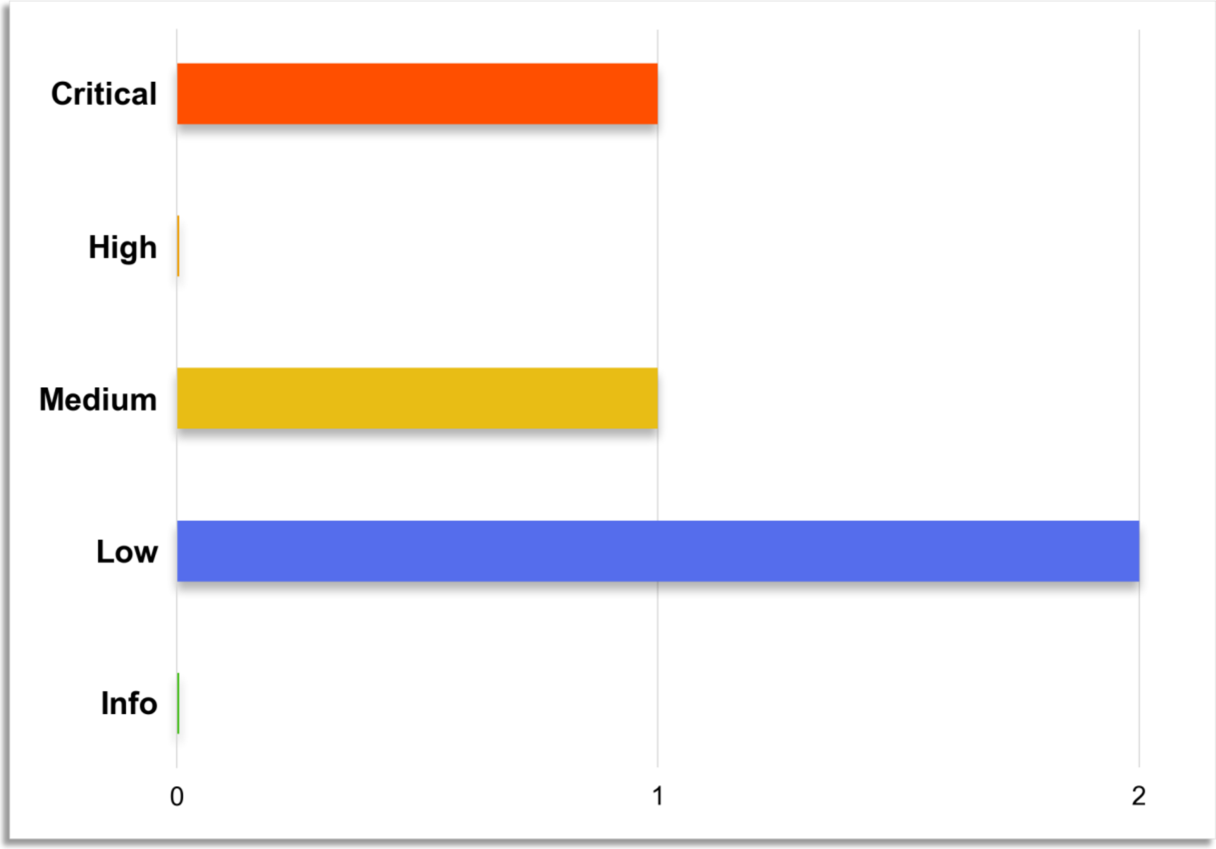

Figure 1: Findings by Severity

FINDINGS

The *Findings* section provides detailed information on each of the findings, including methods of discovery, explanation of severity determination, recommendations, and applicable references.

The following table provides an overview of the findings.

Finding #	Severity	Description
FYEO-LEVANA-01	**Critical**	Attackers can modify the position of other users via the trigger order flow
FYEO-LEVANA-02	**Medium**	Users can remove collateral without impacting notional size
FYEO-LEVANA-03	**Low**	Users funds can be locked when performing operations that accept native tokens
FYEO-LEVANA-04	**Low**	Users incur additional fees when calling removal updates in the CW20 handler

Table 2: Findings Overview

TECHNICAL ANALYSIS

The source code has been manually validated to the extent that the state of the repository allowed. The validation includes confirming that the code correctly implements the intended functionality.

CONCLUSION

Based on our review process, we conclude that the code implements the documented functionality to the extent of the reviewed code.

Technical Findings

General Observations

- The documentation and included slide decks were well written
- The Levana team was very responsive and assisted the auditor with all questions
- There were many invariant checks that prevented unwanted attack scenarios
- There was great test coverage provided in the multi-test files

ATTACKERS CAN MODIFY THE POSITION OF OTHER USERS VIA THE TRIGGER ORDER FLOW

Finding ID: FYEO-LEVANA-01
Severity: **Critical**
Status: **Remediated**

Description

When setting a trigger order, users specify which position ID they want to modify. There are no checks that the position to modify belong to that user.

Proof of Issue

```
#[test]
fn poc_set_other_users_trigger_order_high() {
    // Setup

    let market = PerpsMarket::new(PerpsApp::new_cell().unwrap()).unwrap();

    let trader = market.clone_trader(0).unwrap();

    let cranker = market.clone_trader(1).unwrap();

    // Trader that will execute attack on other trader's positions
    let attacker = market.clone_trader(3).unwrap();

    let take_profit_override =
PriceBaseInQuote::try_from_number(105u128.into()).unwrap();

    let trigger_and_assert = |pos_id: PositionId| {
        market.exec_set_price("105".try_into().unwrap()).unwrap();
        market.exec_crank(&cranker).unwrap();

        let pos = market.query_closed_position(&trader, pos_id).unwrap();
        assert_position_take_profit(&pos).unwrap();
    };

    // Set price of the market to be 1--
    market.exec_set_price("100".try_into().unwrap()).unwrap();
    let (pos_id, _) = market
        .exec_open_position(
            &trader,
            "100",
            "10",
            DirectionToBase::Long,
            "1.0",
            None,
            None,
            None,
        )
        .unwrap();
    // @audit - Supplied the sender to be the attacker address. The attacker was able
to execute a set trigger order for another user.
    market
```

```
        .exec_set_trigger_order(&attacker, pos_id, None, Some(take_profit_override))
        .unwrap();
    // Market price increases and initiates trigger set by the attacker when 
take_profit gains are met
    trigger_and_assert(pos_id);
}
```

a) See coded POC where an attacker can effectively modify others’ positions using a trigger order.

Severity and Impact Summary

This is a critical vulnerability because users can trigger when other users’ positions close. An attacker would essentially be able to close and modify ANY other user’s position without their permission.

Recommendation

The recommendation is to include authority checks when setting trigger orders. Only the owner of a position should be allowed to set their own trigger orders.

USERS CAN REMOVE COLLATERAL WITHOUT IMPACTING NOTIONAL SIZE

Finding ID: FYEO-LEVANA-02

Severity: **Medium**

Status: **Remediated**

Description

When calling `UpdatePositionRemoveCollateralImpactSize`, a user can remove small amounts of collateral without updating the notional size.

Proof of Issue

```
fn update_without_reduction_is_fine() {
    let market = custom_market_setup().unwrap();
    let trader = market.clone_trader(0).unwrap();

    let (pos_id, _) = market
        .exec_open_position(
            &trader,
            "300",
            "10",
            DirectionToBase::Long,
            "1000000.0",
            None,
            None,
            None,
        )
        .unwrap();


    // Move price against us, assert that we have less than 5 collateral.
    market
        .exec_set_price_with_usd("0.95".parse().unwrap(), Some("1".parse().unwrap()))
        .unwrap();

    let pos1 = market.query_position(pos_id).unwrap().notional_size;

    //10^-18
    //market
     //    .exec_update_position_collateral_impact_size(&trader, pos_id, "-
0.000000000000000001".parse().unwrap(), None)
     //    .unwrap();
    //10^-17
    //market
     //    .exec_update_position_collateral_impact_size(&trader, pos_id, "-
0.00000000000000001".parse().unwrap(), None)
     //    .unwrap();
    //10^-16
    market
       .exec_update_position_collateral_impact_size(&trader, pos_id, "-
0.0000000000000001".parse().unwrap(), None)
```

```
        .unwrap();


    let pos2 = market.query_position(pos_id).unwrap().notional_size;
    assert!(market.query_position(pos_id).unwrap().notional_size == "-
2699.999999999999997299".parse().unwrap());
}
```

a) Coded POC displaying no update to `notional_size` when updating the position with `10^-18`, `10^-17`, and `10^-16`. Uncomment each of the examples to show that `pos2` will be the same value for each.

Severity and Impact Summary

Small rounding issues result in incorrect accounting errors that can impact the correct notional size of a market. This severity is ranked as medium due to the high level of effort and amount of capital a user would have to spend on gas fees to perform this attack.

Recommendation

Incorporate checks to verify notional size changes according to the updates made. This would prevent users from removing collateral without effecting notional size.

Users funds can be locked when performing operations that accept native tokens

Finding ID: FYEO-LEVANA-03
Severity: **Low**
Status: **Remediated**

Description

Additional funds sent to any operation that uses `get_native_funds_amount` will be lost and locked into the market contract. This is because funds are sent as an array and only the first item in that array that is a native token is used for the operation.

Proof of Issue

```
pub(crate) fn get_native_funds_amount(
        &self,
        store: &mut dyn Storage,
        info: &MessageInfo,
    ) -> Result<NonZero<Collateral>> {
        let amount = match self.get_token(store)? {
            Token::Native {
                denom,
                decimal_places,
            } => {
                let coin = info
                    .funds
                    .iter()
                    .find(|coin| coin.denom == *denom)
                    .ok_or_else(|| {
                        perp_anyhow!(
                            ErrorId::NativeFunds,
                            ErrorDomain::Market,
                            "no coins attached!"
                        )
                    })?; //@audit - get_native_funds doesn't account for a funds
vector greater than 1
```

Severity and Impact Summary

Users that send any additional funds in the funds array will permanently lose those tokens.

Recommendation

The recommendation is to use the cosmwasm utils for receiving payments. In the code snippet below, the cosmwasm team implements checks for funds array containing multiple items. Alternatively, the team can error out anytime a users sends anything with a funds length greater than 1.

```
/// If exactly one coin was sent, returns it regardless of denom.
/// Returns error if 0 or 2+ coins were sent
```

```
pub fn one_coin(info: &MessageInfo) -> Result<Coin, PaymentError> {
    match info.funds.len() {
        0 => Err(PaymentError::NoFunds {}),
        1 => {
            let coin = &info.funds[0];
            if coin.amount.is_zero() {
                Err(PaymentError::NoFunds {})
            } else {
                Ok(coin.clone())
            }
        }
        _ => Err(PaymentError::MultipleDenoms {}),
    }
}
```

Users incur additional fees when calling removal updates in the CW20 handler

Finding ID: FYEO-LEVANA-04
Severity: **Low**
Status: **Remediated**

Description

`ExecuteMsgs` that don't require sent funds occur in two sections of the market contract code. One of the places theses occur in take place within the CW20 handler. This CW20 handler requires a minimum fee to be paid when executing a removal branch. Essentially, a user would pay extra fees if they were to execute these messages that are nested in the CW20 handler.

Proof of Issue

```
 ExecuteMsg::Receive {
            amount,
            msg,
            sender,
        } => {
            ...
              Token::Cw20 {
                    addr,
                    decimal_places, // @audit-info info.sender is the cw20 token
contract in this case.As long
                } => {
                  ...
                    NonZero::new(Collateral::from_decimal256(Decimal256::from_atomics(
                        amount.u128(),
                        (*decimal_places).into(),
                    )?))
                    .context("Cannot send 0 tokens into the contract")?
                }

                ExecuteMsg::UpdatePositionRemoveCollateralImpactLeverage {...}

                ExecuteMsg::UpdatePositionRemoveCollateralImpactSize {...}

                ExecuteMsg::UpdatePositionLeverage {...}

                ExecuteMsg::UpdatePositionMaxGains {...}

                ExecuteMsg::PlaceLimitOrder {...}
            };
```

Severity and Impact Summary

The impact is low because the fee required is only `amount > 0`. There is also an alternate path for user to execute their actions without having to pay the additional fees.

Recommendation

The recommended action is to encourage users to not use the nested CW20 position update handlers and to use the alternate path. Furthermore, it would be beneficial to remove or force users to use the path that doesn’t require the extra fees.

OUR PROCESS

METHODOLOGY

FYEO Inc. uses the following high-level methodology when approaching engagements. They are broken up into the following phases.

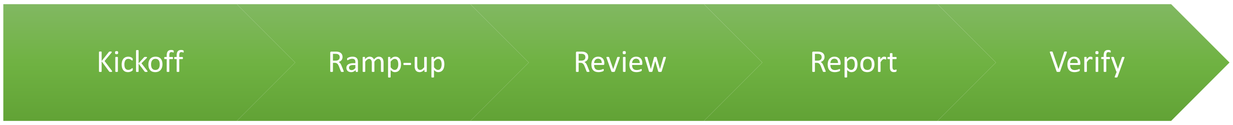

Figure 2: Methodology Flow

KICKOFF

The project is kicked off as the sales process has concluded. We typically set up a kickoff meeting where project stakeholders are gathered to discuss the project as well as the responsibilities of participants. During this meeting we verify the scope of the engagement and discuss the project activities. It's an opportunity for both sides to ask questions and get to know each other. By the end of the kickoff there is an understanding of the following:

- Designated points of contact
- Communication methods and frequency
- Shared documentation
- Code and/or any other artifacts necessary for project success
- Follow-up meeting schedule, such as a technical walkthrough
- Understanding of timeline and duration

RAMP-UP

Ramp-up consists of the activities necessary to gain proficiency on the project. This can include the steps needed for familiarity with the codebase or technological innovation utilized. This may include, but is not limited to:

- Reviewing previous work in the area including academic papers
- Reviewing programming language constructs for specific languages
- Researching common flaws and recent technological advancements

Review

The review phase is where most of the work on the engagement is completed. This is the phase where we analyze the project for flaws and issues that impact the security posture. Depending on the project this may include an analysis of the architecture, a review of the code, and a specification matching to match the architecture to the implemented code.

In this code audit, we performed the following tasks:

1. Security analysis and architecture review of the original protocol
2. Review of the code written for the project
3. Compliance of the code with the provided technical documentation

The review for this project was performed using manual methods and utilizing the experience of the reviewer. No dynamic testing was performed, only the use of custom-built scripts and tools were used to assist the reviewer during the testing. We discuss our methodology in more detail in the following sections.

Code Safety

We analyzed the provided code, checking for issues related to the following categories:

- General code safety and susceptibility to known issues
- Poor coding practices and unsafe behavior
- Leakage of secrets or other sensitive data through memory mismanagement
- Susceptibility to misuse and system errors
- Error management and logging

This list is general and not comprehensive, meant only to give an understanding of the issues we are looking for.

Technical Specification Matching

We analyzed the provided documentation and checked that the code matches the specification. We checked for things such as:

- Proper implementation of the documented protocol phases
- Proper error handling
- Adherence to the protocol logical description

Reporting

FYEO Inc. delivers a draft report that contains an executive summary, technical details, and observations about the project.

The executive summary contains an overview of the engagement including the number of findings as well as a statement about our general risk assessment of the project. We may conclude that the overall risk is low but depending on what was assessed we may conclude that more scrutiny of the project is needed.

We report security issues identified, as well as informational findings for improvement, categorized by the following labels:

- Critical
- High
- Medium
- Low
- Informational

The technical details are aimed more at developers, describing the issues, the severity ranking and recommendations for mitigation.

As we perform the audit, we may identify issues that aren’t security related, but are general best practices and steps that can be taken to lower the attack surface of the project. We will call those out as we encounter them and as time permits.

As an optional step, we can agree on the creation of a public report that can be shared and distributed with a larger audience.

Verify

After the preliminary findings have been delivered, this could be in the form of the approved communication channel or delivery of the draft report, we will verify any fixes within a window of time specified in the project. After the fixes have been verified, we will change the status of the finding in the report from open to remediated.

The output of this phase will be a final report with any mitigated findings noted.

Additional Note

It is important to note that, although we did our best in our analysis, no code audit or assessment is a guarantee of the absence of flaws. Our effort was constrained by resource and time limits along with the scope of the agreement.

While assessing the severity of the findings, we considered the impact, ease of exploitability, and the probability of attack. This is a solid baseline for severity determination.

THE CLASSIFICATION OF VULNERABILITIES

Security vulnerabilities and areas for improvement are weighted into one of several categories using, but is not limited to, the criteria listed below:

Critical – vulnerability will lead to a loss of protected assets

- This is a vulnerability that would lead to immediate loss of protected assets
- The complexity to exploit is low
- The probability of exploit is high

High - vulnerability has potential to lead to a loss of protected assets

- All discrepancies found where there is a security claim made in the documentation that cannot be found in the code
- All mismatches from the stated and actual functionality
- Unprotected key material
- Weak encryption of keys
- Badly generated key materials
- Txn signatures not verified
- Spending of funds through logic errors
- Calculation errors overflows and underflows

Medium - vulnerability hampers the uptime of the system or can lead to other problems

- Insecure calls to third party libraries
- Use of untested or nonstandard or non-peer-reviewed crypto functions
- Program crashes, leaves core dumps or writes sensitive data to log files

Low – vulnerability has a security impact but does not directly affect the protected assets

- Overly complex functions
- Unchecked return values from 3rd party libraries that could alter the execution flow

Informational

- General recommendations