VRust

# Security Assessment

O2Lab VRust Team

28/01/2022 18:03:02

## Contents

## Summary

This report has been prepared for O2Lab VRust Team to discover issues and vulnerabilities in the source code of the O2Lab VRust Team project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques. The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

# Overview

## Project Summary

| | |
|---|---|
| Project Name | O2Lab VRust Team |
| Platform | Ethereum |
| Language | Solana |
| Crate | exchange |
| GitHub Location | https://github.com/parasol-aser/vrust |
| sha256 | Unknown |

## Audit Summary

| | |
|---|---|
| Delivery Date | 28/01/2022 |
| Audit Methodology | Static Analysis |
| Key Components | |

## Vulnerability Summary

| **Vulnerability Level** | **Total** |
|---|---|
| Critical | 46 |
| Major | 0 |
| Medium | 0 |
| Minor | 0 |
| Informational | 0 |
| Discussion | 0 |

## Findings

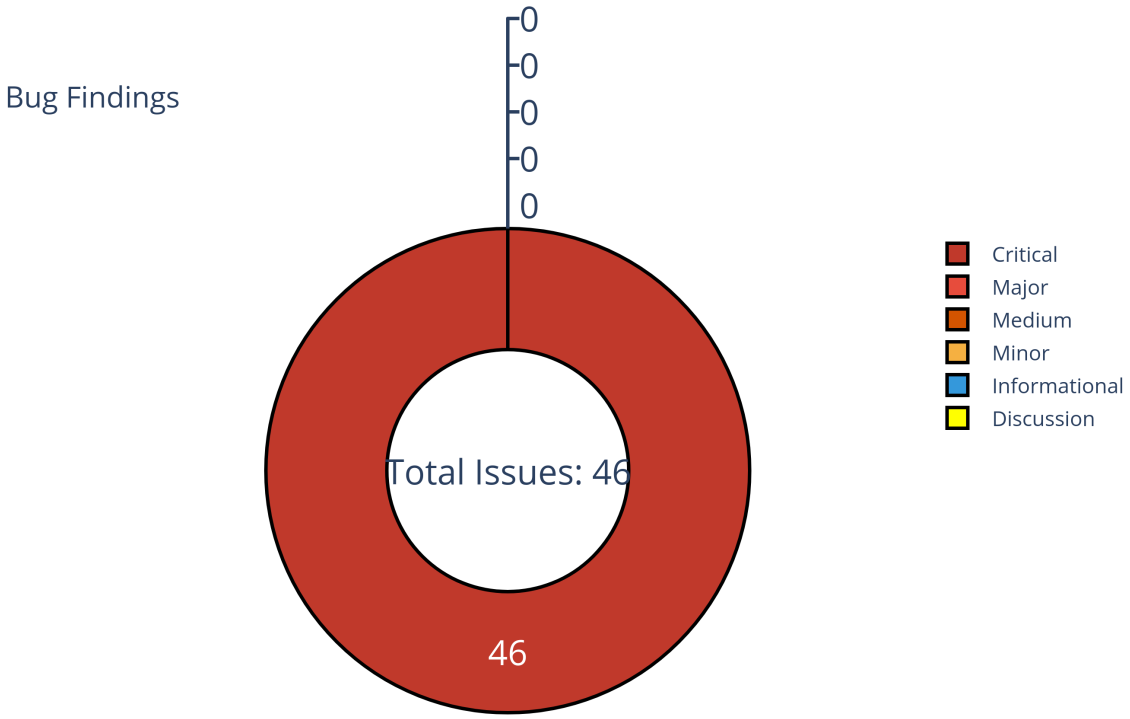

**Figure 1:** Findings

| ID | Title | Category | Severity | Status |
|---|---|---|---|---|
| INT_CVE_0 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_1 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_2 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_3 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_4 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_5 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_6 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_7 | Overflow | Missing Owner Check | Critical | UnResolved |

| ID | Title | Category | Severity | Status |
|---|---|---|---|---|
| INT_CVE_8 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_9 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_10 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_11 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_12 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_13 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_14 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_15 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_16 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_17 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_18 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_19 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_20 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_21 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_22 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_23 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_24 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_25 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_26 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_27 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_28 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_29 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_30 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_31 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_32 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_33 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_34 | Overflow | Missing Owner Check | Critical | UnResolved |

| ID | Title | Category | Severity | Status |
|---|---|---|---|---|
| INT_CVE_35 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_36 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_37 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_38 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_39 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_40 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_41 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_42 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_43 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_44 | Overflow | Missing Owner Check | Critical | UnResolved |
| INT_CVE_45 | Overflow | Missing Owner Check | Critical | UnResolved |

## Issue: INT_CVE_0: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/decimal.rs:78:5: 91:6

```
pub fn to_scale(self, scale: u8) -> Self {
        Self {
            val: if self.scale > scale {
                self.val
                    .checked_div(10u128.pow((self.scale - scale).into()))
                    .unwrap()
            } else {
                self.val
                    .checked_mul(10u128.pow((scale - self.scale).into()))
                    .unwrap()
            },
            scale,
        }
    }

```

- Call Stack

```
programs/exchange/src/decimal.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_1: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:2436:5: 2436:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_2: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:2418:5: 2418:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_3: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:2400:5: 2400:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_4: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:2362:5: 2362:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_5: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:2345:5: 2345:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_6: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:2324:5: 2324:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_7: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1404:5: 1404:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_8: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1364:5: 1364:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_9: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1364:5: 1364:71

```
1364 #[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]
1365
```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_10: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/account.rs:242:5: 245:6

```
pub fn append_collateral(&mut self, new_collateral: Collateral) {
        self.collaterals[(self.head_collaterals) as usize] =
         ↪  new_collateral;
        self.head_collaterals += 1;
    }

```

- Call Stack

```
programs/exchange/src/account.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_11: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1342:5: 1342:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_12: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1342:5: 1342:71

```
1342 #[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]
1343
```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_13: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1275:5: 1275:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_14: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/account.rs:246:5: 249:6

```
pub fn append_synthetic(&mut self, new_synthetic: Synthetic) {
        self.synthetics[(self.head_synthetics) as usize] = new_synthetic;
        self.head_synthetics += 1;
    }

```

- Call Stack

```
programs/exchange/src/account.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_15: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1426:5: 1426:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_16: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1246:5: 1246:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_17: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1201:5: 1201:71

```
1201 #[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]
1202
```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_18: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1188:5: 1188:71

```
1188 #[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]
1189
```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

# Issue: INT_CVE_19: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/decimal.rs:92:5: 105:6

```
pub fn to_scale_up(self, scale: u8) -> Self {
    let decimal = Self::new(self.val, scale);
    if self.scale >= scale {
      decimal.div_up(Self::new(
        10u128.pow((self.scale - scale).try_into().unwrap()),
        0,
      ))
    } else {
      decimal.mul_up(Self::new(
        10u128.pow((scale - self.scale).try_into().unwrap()),
        0,
      ))
    }
  }

```

- Call Stack

```
programs/exchange/src/decimal.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_20: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/account.rs:238:5: 241:6

```
pub fn append_asset(&mut self, new_asset: Asset) {
        self.assets[(self.head_assets) as usize] = new_asset;
        self.head_assets += 1;
    }

```

- Call Stack

```
programs/exchange/src/account.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_21: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:1229:5: 1229:71

```
#[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]

```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_22: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/utils.rs:36:1: 137:2

```rust
pub fn adjust_staking_rounds(state: &mut State, slot: u64) {
    if slot <= state.staking.next_round.start {
        return;
    }
    let slot_diff = slot.checked_sub(state.staking.next_round.start).unwrap();
    let round_diff: u32 = div_up(slot_diff.into(), state.staking.round_length.into())
        .try_into()
        .unwrap();
    match round_diff {
        1 => {
            state.staking.finished_round = state.staking.current_round.clone();
            state.staking.current_round = state.staking.next_round.clone();
            state.staking.next_round = StakingRound {
                start: state
                    .staking
                    .next_round
                    .start
                    .checked_add(state.staking.round_length.into())
                    .unwrap(),
                all_points: state.debt_shares,
                amount: state.staking.amount_per_round,
            }
        }
        2 => {
            state.staking.finished_round = state.staking.next_round.clone();
            state.staking.current_round = StakingRound {
                start: state
```

```
                    .staking
                    .next_round
                    .start
                    .checked_add(state.staking.round_length.into())
                    .unwrap(),
                all_points: state.debt_shares,
                amount: state.staking.amount_per_round,
            };
            state.staking.next_round = StakingRound {
                start: state
                    .staking
                    .next_round
                    .start

                    ↪   .checked_add(state.staking.round_length.checked_mul(2).unwrap().
                    .unwrap(),
                all_points: state.debt_shares,
                amount: state.staking.amount_per_round,
            }
        }
        _ => {
            state.staking.finished_round = StakingRound {
                start: state
                    .staking
                    .next_round
                    .start
                    .checked_add(
                        state
                            .staking
                            .round_length

                            ↪   .checked_mul(round_diff.checked_sub(2).unwrap())
                            .unwrap()
                            .into(),
                    )
                    .unwrap(),
                all_points: state.debt_shares,
                amount: state.staking.amount_per_round,
            };
            state.staking.current_round = StakingRound {
                start: state
                    .staking
```

```
          .next_round
          .start
          .checked_add(
            state
              .staking
              .round_length
              .checked_mul(round_diff.checked_sub(1).unwrap())
              .unwrap()
              .into(),
          )
          .unwrap(),
        all_points: state.debt_shares,
        amount: state.staking.amount_per_round,
      };
      state.staking.next_round = StakingRound {
        start: state
          .staking
          .next_round
          .start
          .checked_add(
            state
              .staking
              .round_length
              .checked_mul(round_diff)
              .unwrap()
              .into(),
          )
          .unwrap(),
        all_points: state.debt_shares,
        amount: state.staking.amount_per_round,
      }
    }
  }
  return;
}
```

- Call Stack

```
programs/exchange/src/utils.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_23: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/utils.rs:138:1: 158:2

```
pub fn adjust_staking_account(exchange_account: &mut ExchangeAccount,
 ↪  staking: &Staking) {
    if exchange_account.user_staking_data.last_update >=
     ↪  staking.current_round.start {
        return;
    } else {
        if exchange_account.user_staking_data.last_update <
         ↪  staking.finished_round.start {
            exchange_account.user_staking_data.finished_round_points =
 ↪  exchange_account.debt_shares;
            exchange_account.user_staking_data.current_round_points =
 ↪  exchange_account.debt_shares;
            exchange_account.user_staking_data.next_round_points =
 ↪  exchange_account.debt_shares;
        } else {
            exchange_account.user_staking_data.finished_round_points =
                exchange_account.user_staking_data.current_round_points;
            exchange_account.user_staking_data.current_round_points =
                exchange_account.user_staking_data.next_round_points;
            exchange_account.user_staking_data.next_round_points =
 ↪  exchange_account.debt_shares;
        }
    }

    exchange_account.user_staking_data.last_update =
        staking.current_round.start.checked_add(1).unwrap();
    return;
}
```

- Call Stack

```
programs/exchange/src/utils.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_24: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/utils.rs:138:1: 158:2

```rust
pub fn adjust_staking_account(exchange_account: &mut ExchangeAccount, staking: &Staking) {
    if exchange_account.user_staking_data.last_update >= staking.current_round.start {
        return;
    } else {
        if exchange_account.user_staking_data.last_update < staking.finished_round.start {
            exchange_account.user_staking_data.finished_round_points = exchange_account.debt_shares;
            exchange_account.user_staking_data.current_round_points = exchange_account.debt_shares;
            exchange_account.user_staking_data.next_round_points = exchange_account.debt_shares;
        } else {
            exchange_account.user_staking_data.finished_round_points =
                exchange_account.user_staking_data.current_round_points;
            exchange_account.user_staking_data.current_round_points =
                exchange_account.user_staking_data.next_round_points;
            exchange_account.user_staking_data.next_round_points = exchange_account.debt_shares;
        }
    }

    exchange_account.user_staking_data.last_update =
        staking.current_round.start.checked_add(1).unwrap();
    return;
}

```

- Call Stack

```
programs/exchange/src/utils.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_25: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/utils.rs:10:1: 26:2

```rust
pub fn check_feed_update(
    assets: &[Asset],
    index_a: usize,
    index_b: usize,
    max_delay: u32,
    slot: u64,
) -> Result<()> {
    // Check assetA
    if assets[index_a].last_update < slot.checked_sub(max_delay.into()).unwrap() {
        return Err(ErrorCode::OutdatedOracle.into());
    }
    // Check assetB
    if assets[index_b].last_update < slot.checked_sub(max_delay.into()).unwrap() {
        return Err(ErrorCode::OutdatedOracle.into());
    }
    return Ok(());
}

```

- Call Stack

```
programs/exchange/src/utils.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_26: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/utils.rs:10:1: 26:2

```rust
pub fn check_feed_update(
    assets: &[Asset],
    index_a: usize,
    index_b: usize,
    max_delay: u32,
    slot: u64,
) -> Result<()> {
    // Check assetA
    if assets[index_a].last_update < slot.checked_sub(max_delay.into()).unwrap() {
        return Err(ErrorCode::OutdatedOracle.into());
    }
    // Check assetB
    if assets[index_b].last_update < slot.checked_sub(max_delay.into()).unwrap() {
        return Err(ErrorCode::OutdatedOracle.into());
    }
    return Ok(());
}

```

- Call Stack

```
programs/exchange/src/utils.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_27: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```
151 pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
152     // decimals of token = 6
153     let one_sny = Decimal::from_integer(1).to_sny().val;
154     let amount = sny_amount.val;
155 
156     let v: u16 = match () {
157         () if amount < one_sny * 100 => 0,
158         () if amount < one_sny * 200 => 1,
159         () if amount < one_sny * 500 => 2,
160         () if amount < one_sny * 1_000 => 3,
161         () if amount < one_sny * 2_000 => 4,
162         () if amount < one_sny * 5_000 => 5,
163         () if amount < one_sny * 10_000 => 6,
164         () if amount < one_sny * 25_000 => 7,
165         () if amount < one_sny * 50_000 => 8,
166         () if amount < one_sny * 100_000 => 9,
167         () if amount < one_sny * 250_000 => 10,
168         () if amount < one_sny * 250_000 => 10,
169         () if amount < one_sny * 500_000 => 11,
170         () if amount < one_sny * 1_000_000 => 12,
171         () if amount < one_sny * 2_000_000 => 13,
172         () if amount < one_sny * 5_000_000 => 14,
173         () if amount < one_sny * 10_000_000 => 15,
174         () => 15,
175     };
176     return Decimal::from_percent(v);
177 }
178 
```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_28: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_29: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_30: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_31: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_32: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_33: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_34: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_35: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_36: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_37: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_38: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_39: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_40: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```
151 pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
152     // decimals of token = 6
153     let one_sny = Decimal::from_integer(1).to_sny().val;
154     let amount = sny_amount.val;
155 
156     let v: u16 = match () {
157         () if amount < one_sny * 100 => 0,
158         () if amount < one_sny * 200 => 1,
159         () if amount < one_sny * 500 => 2,
160         () if amount < one_sny * 1_000 => 3,
161         () if amount < one_sny * 2_000 => 4,
162         () if amount < one_sny * 5_000 => 5,
163         () if amount < one_sny * 10_000 => 6,
164         () if amount < one_sny * 25_000 => 7,
165         () if amount < one_sny * 50_000 => 8,
166         () if amount < one_sny * 100_000 => 9,
167         () if amount < one_sny * 250_000 => 10,
168         () if amount < one_sny * 250_000 => 10,
169         () if amount < one_sny * 500_000 => 11,
170         () if amount < one_sny * 1_000_000 => 12,
171         () if amount < one_sny * 2_000_000 => 13,
172         () if amount < one_sny * 5_000_000 => 14,
173         () if amount < one_sny * 10_000_000 => 15,
174         () => 15,
175     };
176     return Decimal::from_percent(v);
177 }
178 
```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_41: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_42: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_43: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/math.rs:151:1: 177:2

```rust
pub fn amount_to_discount(sny_amount: Decimal) -> Decimal {
    // decimals of token = 6
    let one_sny = Decimal::from_integer(1).to_sny().val;
    let amount = sny_amount.val;

    let v: u16 = match () {
        () if amount < one_sny * 100 => 0,
        () if amount < one_sny * 200 => 1,
        () if amount < one_sny * 500 => 2,
        () if amount < one_sny * 1_000 => 3,
        () if amount < one_sny * 2_000 => 4,
        () if amount < one_sny * 5_000 => 5,
        () if amount < one_sny * 10_000 => 6,
        () if amount < one_sny * 25_000 => 7,
        () if amount < one_sny * 50_000 => 8,
        () if amount < one_sny * 100_000 => 9,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 250_000 => 10,
        () if amount < one_sny * 500_000 => 11,
        () if amount < one_sny * 1_000_000 => 12,
        () if amount < one_sny * 2_000_000 => 13,
        () if amount < one_sny * 5_000_000 => 14,
        () if amount < one_sny * 10_000_000 => 15,
        () => 15,
    };
    return Decimal::from_percent(v);
}

```

- Call Stack

```
programs/exchange/src/math.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_44: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/account.rs:178:5: 181:6

```
pub fn append(&mut self, entry: CollateralEntry) {
        self.collaterals[(self.head) as usize] = entry;
        self.head += 1;
    }

```

- Call Stack

```
programs/exchange/src/account.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

## Issue: INT_CVE_45: IntegerCve - Overflow

| Category | Severity | Status |
|---|---|---|
| Missing Owner Check | Critical | UnResolved |

- Location

programs/exchange/src/lib.rs:2383:5: 2383:71

```
2383 #[access_control(admin(&ctx.accounts.state, &ctx.accounts.admin))]
2384
```

- Call Stack

```
programs/exchange/src/lib.rs
```

- description:

Description of the bug here.

- link:

GitHub Link to be added.

- alleviation:

Some alleviation steps here.

# Appendix

Copied from https://leaderboard.certik.io/projects/aave

## Finding Categories

### Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

### Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

### Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

### Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

### Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

## Checksum Calculation Method

The “Checksum” field in the “Audit Scope” section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux “sha256sum” command against the target file.

## Disclaimer

Copied from https://leaderboard.certik.io/projects/aave

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.