Security Audit Report

Coalescefi

Date:	2025-12-02
Audit Commit:	`9277a7996fb89630439ec5174b42a9da60803857`
Audit Period:	2025-11-15 - 2025-11-24
Total Findings:	10

Executive Summary

The following report details the findings from the security audit.

Severity	Count
CRITICAL	0
HIGH	2
MEDIUM	7
LOW	1
INFORMATIONAL	0

Findings Log

#1 HIGH

Incorrect use of `fee_receiver`

DESCRIPTION

The `process_repay_obligation_liquidity` instruction incorrectly uses `platform_config.fee_receiver` as a token account in the `penalty_amount > 0` scenario. This causes early/late repayment to fail.

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L3240

MITIGATION SUGGESTION

Derive the ATA for the `config.fee_receiver` address and require the caller to supply the account.

REMEDIATION

Fixed in commit `0873ff94b6614e28be38adcf0d9457a18521aa82`

#2 HIGH

Attributed borrow values not updated on repayment can lead to DoS

DESCRIPTION

Repaying an obligation never updates the reserve or collateral `attributed_borrow_value`, so the reserve remains stays at its previous limit even when `borrowed_amount_wads` falls to zero. This leaves the entire pool unable to accept new borrows despite having full capacity.

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L2465

MITIGATION SUGGESTION

After settling a borrow in `process_repay_obligation_liquidity`, recompute collateral and reserve attribution via `update_borrow_attribution_values`, similar to the borrow and withdraw paths.

REMEDIATION

Fixed in commit `f41ca369519a1dc9c125a6e5ad89f11d613a521e`

#3 MEDIUM

1-lamport DoS in PDA creation

DESCRIPTION

`create_pda_account` creates PDAs with a single system_instruction::create_account CPI and assumes the PDA has zero lamports. This makes it vulnerable to 1 lamport DoS

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/coalescefi_processor.rs#L462

MITIGATION SUGGESTION

Handle the prefunded PDA case with something similar to:

```
        if account_lamports != 0 && account_to_create.data_is_empty() {
            let defund_created_account = system_instruction::transfer(
                account_to_create.key,
                fee_payer.key,
                account_lamports,
            );
            invoke_signed(
                &defund_created_account,
                &[
                    system_program.clone(),
                    fee_payer.clone(),
                    account_to_create.clone(),
                ],
                &[signer_seeds],
            )?;
        }
```

REMEDIATION

Fixed in commit `02abfd4e5a9a784b52b718a4252e1c37be320ef5`

#4 MEDIUM

Stale token limits persist when access mode changes

DESCRIPTION

Switching a user out of `ACCESS_MODE_TOKEN_SPECIFIC` only zeroes `tracked_tokens_count` and never clears `token_limits`. When an admin toggles `TOKEN_SPECIFIC` → `UNLIMITED` → `TOKEN_SPECIFIC`, every slot in `token_limits` still holds the prior `TokenLimit` entries. Once token-specific mode is re-enabled:

- `can_borrow` continues to enforce those stale limits, so the user retains borrow permissions the admin intended to remove.
- Any attempt to add a new mint limit fails with `LendingError::MaxTokenTypesExceeded`, even though `tracked_tokens_count` reports 0. Admins cannot configure new limits and users keep unintended access.

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/coalescefi_state/user_state.rs#L307

MITIGATION SUGGESTION

When leaving `ACCESS_MODE_TOKEN_SPECIFIC`, clear every entry in `token_limits` (set them to `None`) . Restrict `add_token_limit` to succeed only while `access_mode == ACCESS_MODE_TOKEN_SPECIFIC`.

REMEDIATION

Fixed in commit `3fb5eb673f70fd9236113a08499a4a1fea74f598`

#6 MEDIUM

Incorrect `PlatformState.pool_reserve`

DESCRIPTION

The legacy branch of `process_init_reserve` incorrectly stores `lending_market_info.key` in `platform_config.pool_reserve` instead of `reserve_info.key`.

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L814

MITIGATION SUGGESTION

When `pda_type` is `Legacy`, bind `platform_config.pool_reserve` to `reserve_info.key` on first initialization.

REMEDIATION

Fixed in commit `ff019645bf1d940c7bb835d2237c4edff900832e`

#7 MEDIUM

Fixed-term pool expiry bypass via missing `PoolRateConfig`

DESCRIPTION

Deposits into fixed-term pools only check expiry if the optional `PoolRateConfig` PDA is supplied in `remaining_accounts`. When the PDA is omitted, `load_pool_rate_config` returns `None`, making `check_pool_expiry` a no-op and allowing deposits into already-expired pools.

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L238

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L301

MITIGATION SUGGESTION

Return an error when `PoolRateConfig` PDA is missing

REMEDIATION

Fixed in commit `5b80177ca6a4b9a3aa4e300cf418bb7ec377f016`

#8 MEDIUM

Lack of PDA validation

`process_modify_user_reserve_access` & `process_modify_user_access`

DESCRIPTION

The instructions `process_modify_user_access` and `process_modify_user_reserve_access` in take a target `user` but fail to validate that the `UserState` account is the correct PDA. Since this is an admin instruction it would only lead to issues if the admin were to make a mistake.

LOCATION

- https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/coalescefi_processor.rs#L330
- https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/coalescefi_processor.rs#L408

MITIGATION SUGGESTION

Add an account key check.

REMEDIATION

Fixed in commit `13ab4349468b2afd6578a543ced3fd064536716e`

#9 MEDIUM

Excessive penalties can be charged during early Repayment

DESCRIPTION

During `process_repay_obligation_liquidity`, the early/late fixed-term penalty is calculated directly from the `liquidity_amount` instruction input before the actual settlement amount is known. If a user supplies a liquidity amount larger than the outstanding fixed-term debt, `calculate_repay` later caps the principal transfer to the real debt, but the penalty remains tied to the inflated input.

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L3117

MITIGATION SUGGESTION

Defer penalty calculation until after `calculate_repay` returns and base it on the actual `repay_amount`

REMEDIATION

Fixed in commit `d8cf39667a43ead3d8e0be442e2405880a8d8501`

#10 MEDIUM

Borrow/repay token-specific `UserState` mismatch blocks repayment

DESCRIPTION

When token-specific access is configured both globally and per-reserve, `borrow_obligation_liquidity` increments only the per-reserve `UserState`, `but repay_obligation_liquidity` decrements only the global `UserState`. The global entry stays at `borrowed=0`, so `remove_borrowed` rejects any positive repay with `InvalidAmount`.

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L2875

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L3174

MITIGATION SUGGESTION

Use consistent scope for borrow/repay token tracking: either update both global and per-reserve entries, or prefer per-reserve when present and fall back to global otherwise. In repay, attempt per-reserve first (matching borrow), then global; or mirror the chosen scope in both directions.

REMEDIATION

Fixed in commit `821870ccc0dda6f1c12e7a45c3edd9f673f5aa88`

#11 LOW

Blacklisted user can still deposit obligation collateral

DESCRIPTION

`DepositObligationCollateral` accepts deposits without validating the caller's `UserState`, so globally blacklisted users can continue increasing their collateral positions and protocol exposure. Withdrawals do enforce the blacklist, leading to inconsistent access control.

LOCATION

https://github.com/saguarocrypto/coalescefi-program/blob/9277a7996fb89630439ec5174b42a9da60803857/solend-coalescefi/token-lending/program/src/processor.rs#L2017

MITIGATION SUGGESTION

Require the global `UserState` PDA in `process_deposit_obligation_collateral` (and the combined deposit-liquidity+collateral path) and reject when `is_blacklisted`, mirroring the withdrawal flow.

REMEDIATION

Fixed in commit `3989fd933192e2a7ca57b9fb5a2152c8ba79d09a`