ClarityAlliance

BITFLOW STABLESWAP MIDPOINT (UPGRADE) SECURITY REVIEW

Conducted by:
KRISTIAN APOSTOLOV, ALIN BARBATEI (ABA)

APRIL 25TH, 2025

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

1. About Clarity Alliance

Clarity Alliance is a team of expert whitehat hackers specialising in securing protocols on Stacks.

They have disclosed vulnerabilities that have saved millions in live TVL and conducted thorough reviews for some of the largest projects across the Stacks ecosystem.

Learn more about Clarity Alliance at clarityalliance.org.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

2. Disclaimer

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 Clarity Alliance 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. Clarity Alliance’s position is that each company and individual are responsible for their own due diligence and continuous security. Clarity Alliance’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 Clarity Alliance are 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. Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract’s safety and security. Therefore, Clarity Alliance does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

3. Introduction

A time-boxed security review of Bitflow StableSwap Midpoint (Upgrade), where Clarity Alliance reviewed the scope and provided insights on improving the protocol.

4. About Bitflow StableSwap

Bitflow StableSwap is the first protocol designed to enable users to efficiently swap stable assets, including stablecoins, within the Bitcoin ecosystem. It operates on the Stacks layer, a platform specifically designed to facilitate smart contracts and decentralized applications on Bitcoin.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

5. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

5.1 Impact

- High - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- Medium - only a small amount of funds can be lost (such as leakage of value) or a core functionality of the protocol is affected.
- Low - can lead to any kind of unexpected behavior with some of the protocol's functionalities that's not so critical.

5.2 Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.
- Medium - only a conditionally incentivized attack vector, but still relatively likely.
- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

5.3 Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

6. Security Assessment Summary

Scope

The following contracts were in the scope of the security review:

- `contracts/stableswap-pool-trait-v-1-1.clar`
- `contracts/stableswap-emissions-stx-ststx-stx-v-1-1.clar`
- `contracts/stableswap-core-v-1-1.clar`
- `contracts/stableswap-staking-stx-ststx-v-1-1.clar`
- `contracts/stableswap-swap-helper-v-1-1.clar`
- `contracts/token-stx-v-1-1.clar`
- `contracts/sip-010-trait-ft-standard-v1-1-1.clar`
- `contracts/stableswap-pool-stx-ststx-v1-1-1.clar`

Initial Commit Reviewed:
6f0518a2065f3fb24a41d09982836db0dfc77c6f

Final Commit After Audit Remediations:
9fc49883c59a482ce8e22a5ea4a5ea410af7bbfa

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don’t Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

7. Executive Summary

Over the course of the security review, Kristian Apostolov, Alin Barbatei (ABA) engaged with - to review Bitflow StableSwap. In this period of time a total of **11** issues were uncovered.

Protocol Summary

Protocol Name	Bitflow StableSwap
Date	April 25th, 2025

Findings Count

Severity	Amount
High	2
Medium	1
Low	4
QA	4
Total Findings	**11**

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

Summary of Findings

ID	Title	Severity	Status
[H-01]	Imbalanced Withdrawals Return Incorrect Token Amounts	High	Resolved
[H-02]	Imbalanced Withdrawals Don't Remove Fees From Pool Balance	High	Resolved
[M-01]	Cooldown Granularity May Block Withdrawals with Volatile Pairs	Medium	Resolved
[L-01]	Excessive Fees with withdraw-imbalanced-liquidity	Low	Acknowledged
[L-02]	Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals	Low	Resolved
[L-03]	Inconsistent Checks Between Previewing LPs and Adding Liquidity	Low	Resolved
[L-04]	Imbalanced Pools Can Be Created Without Using Midpoint	Low	Resolved
[QA-01]	set-freeze-midpoint-manager Can Be Called Multiple Times	QA	Resolved
[QA-02]	withdraw-imbalanced-liquidity Misleading Internal Variable Name	QA	Resolved
[QA-03]	Redundant Return Amounts in withdraw-imbalanced-liquidity	QA	Resolved
[QA-04]	Typographical Error	QA	Resolved

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

8. Findings

8.1. High Findings

[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts

Description

The new `withdraw-imbalanced-liquidity` function in the `stableswap-core-v-1-1` contract enables the withdrawal of X and Y tokens from the pool by specifying the desired amounts and a maximum LP amount to be burned to retrieve them.

During the calculation of the updated amounts, the contract applies a liquidity fee. This fee is scaled for precision before being deducted from the intended user amount.

However, the liquidity fee (scaled) is subtracted from the non-scaled input amounts for both X and Y tokens:

```
(updated-x-amount-scaled (- x-amount x-amount-fee-liquidity-scaled))
(updated-y-amount-scaled (- y-amount y-amount-fee-liquidity-scaled))
```

This results in an incorrect amount of tokens being returned to users and may even lead to blocked withdrawals, as the `x-amount-fee-liquidity-scaled` can exceed the amount itself. The fee was capped to the scaled X and Y token amounts, not to the non-scaled versions.

Recommendation

When calculating the `updated-x-amount-scaled` and `updated-y-amount-scaled` in the `withdraw-imbalanced-liquidity` function, deduct the scaled fee from the scaled amounts (`x-amount-scaled` and `y-amount-scaled`) instead of the non-scaled versions.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance

Description

The new `withdraw-imbalanced-liquidity` function in the `stableswap-core-v-1-1` contract allows users to withdraw X and Y tokens from the pool by specifying the desired amounts and a maximum LP amount to be burned to retrieve them.

During the calculation of the updated amounts, the contract applies a liquidity fee, which is a percentage deducted from both the X and Y tokens.

The current implementation mistakenly does not subtract the fee amount from the final updated balance, resulting in a ghost amount and unbacked token amount in the pools.

This issue arises because the new balance, `updated-balance-[x/y]-post-fee-scaled` , only deducts the amount going to the user (`updated-[x/y]-amount-scaled`), which excludes the fees, instead of the entire amount removed.

```
(updated-x-amount-scaled (- x-amount x-amount-fee-liquidity-scaled))
(updated-y-amount-scaled (- y-amount y-amount-fee-liquidity-scaled))

;; ... code ...
(updated-balance-x-post-fee-scaled (- x-balance-scaled updated-x-amount-scaled))
(updated-balance-y-post-fee-scaled (- y-balance-scaled updated-y-amount-scaled))

;; ... code ...
(updated-pool-balances-post-fee
(scale-down-amounts updated-balance-x-post-fee-scaled updated-balance-y-post-fee-sca
(updated-x-balance-post-fee (get x-amount updated-pool-balances-post-fee))
(updated-y-balance-post-fee (get y-amount updated-pool-balances-post-fee))

;; ... code ...
;; Update pool balances and d value
(try!
(contract-call? pool-trait update-pool-balances updated-x-balance-post-fee updated-y
```

If the fees remained in the pool, this would not be an issue. However, since the fees are sent to the `fee-address` , they must also be removed from the pool's internal accounting.

```
;; Transfer x-amount-fees-liquidity x tokens from pool contract to fee-address
(if (> x-amount-fees-liquidity u0)
(try!
(contract-call? pool-trait pool-transfer x-token-trait x-amount-fees-liquidity fee
false
)

;; Transfer y-amount-fees-liquidity y tokens from pool contract to fee-address
(if (> y-amount-fees-liquidity u0)
(try!
(contract-call? pool-trait pool-transfer y-token-trait y-amount-fees-liquidity fee
false
)
```

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don’t Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

Description

When calculating the `updated-balance-[x/y]-post-fee-scaled` variables, subtract the `[x/y]-amount-scaled` variable, instead of the `updated-[x/y]-amount-scaled` (which has the fees subtracted).

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

8.2. Medium Findings

[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs

Description

The latest codebase commit introduced an option to specify a cooldown period during which no LP withdrawals can occur. This cooldown is measured from the last update of the midpoint, in terms of passed burn blocks.

Since Bitcoin burn blocks are generally mined every 10 minutes, even a 1-block cooldown can effectively block withdrawals when the pool consists of volatile pairs.

Consider the following scenario:

- The STX/stSTX ratio is initially set at 1.2.
- Due to volatile market conditions or other events, the STX/stSTX ratio fluctuates dramatically between 1.2 and 1.5.
- In such cases, the midpoint manager may need to update the ratio more frequently, possibly more than once every 10 minutes.
- If the ratio is updated within 10-minute intervals, even with a minimum cooldown of 1 burn block, users would be unable to exchange their LP until conditions stabilize. This situation forces the team to either remove the cooldown or prevent users from withdrawing during this period.

Recommendation

Utilize `stacks-block-height` instead of burn block height to evaluate the withdrawal cooldown. This approach allows for implementing a finer, time-equivalent granularity as needed.

CONTENTS
1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

8.3. Low Findings

[L-01] Excessive Fees with `withdraw-imbalanced-liquidity`

Description

The new `withdraw-imbalanced-liquidity` function in the `stableswap-core-v-1-1` contract enables users to withdraw X and Y tokens from the pool by specifying the desired amounts and a maximum LP amount to be burned for retrieval.

From a fee perspective, users previously encountered:

- Fees on swaps
- Fees on adding liquidity (via `add-liquidity`)
- No fees on withdrawing liquidity (via `withdraw-proportional-liquidity`)

However, the `withdraw-imbalanced-liquidity` function also imposes a liquidity fee.

This additional fee may discourage LP holders from using this function, as its counterpart, `withdraw-proportional-liquidity` , does not impose any fee.

Recommendation

If the fee for `withdraw-imbalanced-liquidity` is intentional, acknowledge this issue; otherwise, consider removing the fee entirely. Note: if the imbalanced-withdraw fee is desired, it is appropriate to apply it to both token amounts.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals

Description

The new `withdraw-imbalanced-liquidity` function in the `stableswap-core-v-1-1` contract allows users to withdraw X and Y tokens from the pool by specifying the desired amounts and the maximum LP amount to be burned to obtain them.

Due to rounding down, the LP amount burned from the user will be less than optimal.

```
(dlp (/ (* total-shares (- d-a updated-d)) d-a))
```

Continuous rounding that benefits users may, over time, skew the pool balances and lead to unforeseen issues.

Description

Calculate the `dlp` variable using a rounded UP division instead of rounding down.

Note: This behavior is also present in the original Curve StableSwap implementation.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity

Description

When a user previews the amount of LPs they would receive for a given amount of tokens supplied, the `get-dlp` function is used. However, when a user actually supplies liquidity, the `add-liquidity` function from the core contract is used.

The `add-liquidity` function includes a check to ensure that the newly added `X` and `Y` token amounts are less than ten times the existing balances of the pool.

```
;;
   Assert that x-amount and y-amount are less than 10 times x-balance and y-balance
(asserts! (< x-amount
   (* x-balance MAX_AMOUNT_PER_BALANCE_MULTIPLIER)) ERR_INVALID_AMOUNT)
(asserts! (< y-amount
   (* y-balance MAX_AMOUNT_PER_BALANCE_MULTIPLIER)) ERR_INVALID_AMOUNT)
```

However, the `get-dlp` function does not perform these checks.

Third-party integrators might mistakenly assume that the return value from the `get-dlp` function call is valid, while the actual call could revert, leading to minor integration issues.

Recommendation

Incorporate the value–balance validation checks present in the `add-liquidity` function into the `get-dlp` function as well.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

[L-04] Imbalanced Pools Can Be Created Without Using Midpoint

Description

Before the introduction of the midpoint option, creating a pool via the `create-pool` function in the `stableswap-core-v-1-1` contract required that the provided `X` and `Y` token balances be equal.

With the addition of the midpoint option, this requirement was removed. However, if a pool is created without using the midpoint option, there is no check to ensure that the balances are equal.

This oversight allows for the accidental creation of imbalanced pools from the outset when the midpoint is not utilized.

Recommendation

In the `stableswap-core-v-1-1::create-pool` function, ensure that the `X` and `Y` token balances (scaled) are equal if no midpoint configuration is applied.

Example implementation:

```
(define-constant ERR_UNEQUAL_POOL_BALANCES (err u1032))

;; Assert that balances are equal if midpoint is not used
(if (and
       (is-eq midpoint-primary-numerator midpoint-primary-denominator)
       (is-eq midpoint-withdraw-numerator midpoint-withdraw-denominator))
   (asserts! (is-eq x-balance-scaled y-balance-scaled) ERR_UNEQUAL_POOL_BALANCES)
   false
)
```

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

8.4. QA Findings

[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times

Description

Admins have the ability to freeze the midpoint manager using the `stableswap-core-v-1-1::set-freeze-midpoint-manager` function. Once this function is executed, the pool remains permanently frozen with the same principal manager.

However, the current implementation permits admins to repeatedly call `set-freeze-midpoint-manager`, resulting in event spamming, even though no additional side effects occur. This behavior can create problems for off-chain monitoring systems that parse the `“set-freeze-midpoint-manager”` event.

Recommendation

Prevent the `set-freeze-midpoint-manager` function from being called on a pool that is already frozen.

CONTENTS
1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name

Description

In the `stableswap-core-v-1-1::withdraw-imbalanced-liquidity` function, the temporary tuple variable used to store the scaled input amounts is named `amounts-added-scaled`. This is misleading because these amounts are actually being removed from the pool, not added.

Recommendation

Rename the `amounts-added-scaled` variable to `amounts-removed-scaled` in the `withdraw-imbalanced-liquidity` core function.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

[QA-03] Redundant Return Amounts in `withdraw-imbalanced-liquidity`

Description

In the `stableswap-core-v-1-1::withdraw-imbalanced-liquidity` function, the returned values are `(ok {x-amount: updated-x-amount-scaled, y-amount: updated-y-amount-scaled, dlp: dlp})`.

The `updated-x-amount-scaled` and `updated-y-amount-scaled` values are not used externally, as they are scaled amounts intended for internal use. Additionally, the function is invoked with non-scaled amounts, which diminishes the significance of returning these values.

Recommendation

Modify the return statement in the `withdraw-imbalanced-liquidity` function to include only `(ok dlp)`.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Bitflow StableSwap 4
5. Risk Classification 5
5.1. Impact 5
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 6
7. Executive Summary 7
8. Summary of Findings 8
8.1. High Findings 9
[H-01] Imbalanced Withdrawals Return Incorrect Token Amounts 9
[H-02] Imbalanced Withdrawals Don't Remove Fees From Pool Balance 10
8.2. Medium Findings 12
[M-01] Cooldown Granularity May Block Withdrawals with Volatile Pairs 12
8.3. Low Findings 13
[L-01] Excessive Fees with withdraw-imbalanced-liquidity 13
[L-02] Burned LP Rounding Is Against The Protocol For Imbalanced Withdrawals 14
[L-03] Inconsistent Checks Between Previewing LPs and Adding Liquidity 15
[L-04] Imbalanced Pools Can Be Created Without Using Midpoint 16
8.4. QA Findings 17
[QA-01] set-freeze-midpoint-manager Can Be Called Multiple Times 17
[QA-02] withdraw-imbalanced-liquidity Misleading Internal Variable Name 18
[QA-03] Redundant Return Amounts in withdraw-imbalanced-liquidity 19
[QA-04] Typographical Error 20

[QA-04] Typographical Error

Description

In the `stableswap-core-v-1-1` contract, there is a typographical error in the comment describing the `global-imbalanced-withdraws` flag. The word `imabalanced` is misspelled and should be corrected to `imbalanced`.

Recommendation

Correct the identified typo.