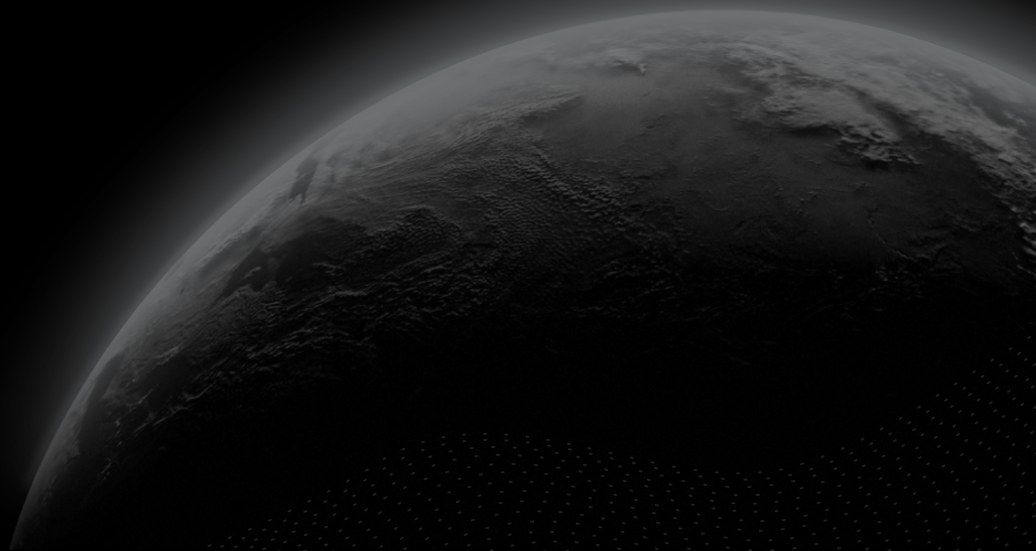




Security Assessment

Minswap - AMM DEX V2

CertiK Assessed on Mar 1st, 2024





Certik Assessed on Mar 1st, 2024

Minswap - AMM DEX V2

The security assessment was prepared by Certik, the leader in Web3.0 security.

Executive Summary

TYPES

Trading-AMM

ECOSYSTEM

Cardano (ADA)

METHODS

Manual Review, Static Analysis

LANGUAGE

Aiken

TIMELINE

Delivered on 03/01/2024

KEY COMPONENTS

N/A


CODEBASE

<https://github.com/minswap/amm-dex-v2/>[View All in Codebase Page](#)

COMMITTS

[17ab42bcde1513a1138e3124906a22c564588c37](#)[d51628e907f4c2a5b4f55f95c58a2a15bed066ef](#)[ec38cd102e6bdb6b8a2bfd489f850d0ffb97ec01](#)[View All in Codebase Page](#)

Highlighted Centralization Risks

 Fees are bounded by 10%

Vulnerability Summary



10

Total Findings

8

Resolved

0

Mitigated

0

Partially Resolved

2

Acknowledged

0

Declined

 0 Critical

Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.

 3 Major

2 Resolved, 1 Acknowledged



Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.

 1 Medium

1 Resolved



Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.

 1 Minor

1 Resolved



Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.

 5 Informational

4 Resolved, 1 Acknowledged



Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | MINSWAP - AMM DEX V2

I **Summary**

[Executive Summary](#)

[Vulnerability Summary](#)

[Codebase](#)

[Audit Scope](#)

[Approach & Methods](#)

I **Dependencies**

[Out Of Scope Dependencies](#)

[Assumptions](#)

[Recommendations](#)

I **Findings**

[FAC-01 : Creation of Pools with Invalid Parameters](#)

[MIN-01 : Logical issue in fee settings](#)

[VAL-01 : Centralization Related Risks](#)

[FAC-02 : Pool Creation Allows Complete Asset Withdrawal](#)

[ORD-02 : Missing Check on `io_ratio_denominator`](#)

[AUT-01 : Incorrect comment](#)

[GLOBAL-01 : Unit Test Documentation](#)

[MIN-02 : `TODO` Comments](#)

[ORD-01 : Missing Formulas for `WithdrawImbalance` and `PartialSwap`](#)

[ORD-03 : Typos](#)

I **Optimizations**

[MAT-01 : Potential Optimization in `math.calculate_withdraw_imbalance\(\)`](#)

I **Appendix**

I **Disclaimer**

CODEBASE | MINSWAP - AMM DEX V2

Repository









<https://github.com/minswap/amm-dex-v2/>


Commit

[17ab42bcde1513a1138e3124906a22c564588c37d51628e907f4c2a5b4f55f95c58a2a15bed066efec38cd102e6bdb6b8a2bfd489f850d0ffb97ec01242e522569305bd3fb5fac2d56079728f97faf6a058d9f37657a30fae31895dd49f962e40af2a0add9680488517bceb186ebc8d0ebd257c787e2b659f1ac03dee95d9a3a4981f655ec5fc51987e87a9](#)

AUDIT SCOPE | MINSWAP - AMM DEX V2

9 files audited ● 3 files with Acknowledged findings ● 3 files with Resolved findings ● 3 files without findings

ID	Repo	File	SHA256 Checksum
● MAT	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ lib/amm_dex_v2/math.ak	88cd180ac4f6b23219eda6b2dc9551cc73e 7807b56eaf98d080d9336cd127e21
● FAC	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ validators/factory_validator.ak	42b0e3db45bc0742c9108115ae9589b3b99 972a199aff2bfda87345417e6b39c
● POL	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ validators/pool_validator.ak	ea16dd0eee72454a8617f089679ff0353231 e4ddb24309dc13d9505c1c6785d8
● POO	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ lib/amm_dex_v2/pool_validation.a k	ffa625bbb7ffb1cfb6d3c0ba634ebde4cce0 414e478435903a822f2572fb4cb
● ORD	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ lib/amm_dex_v2/order_validation. ak	07915281e39228a4156b1608b6eebd38f65 82356e239538a2ad2fa68cb0cc9b8
● AUT	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ validators/authen_minting_policy. ak	3d15177ddcb962814c137025eeb6a45c8bc 439a782637a937b0605c497226813
● CKP	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ lib/amm_dex_v2/utils.ak	cc3339289c479a18365c1ef2e38d284889fc a077dc5e04d13a131211bf2fc37b
● TYP	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ lib/amm_dex_v2/types.ak	28e8ed08efc01fb26586f7fda88d6b42f508a 5798524870d7571f6113c6e6895

ID	Repo	File	SHA256 Checksum
● ORE	CertiKProject/certik-audit-projects	 minswap-dex-v2-17ab42bcde151 3a1138e3124906a22c564588c37/ validators/order_validator.ak	cdef3dc4110bf0a07e207f5b7fbb06f2f2b5f7 730f212a33c03677211bfc234c

APPROACH & METHODS | MINSWAP - AMM DEX V2

This report has been prepared for Minswap to discover issues and vulnerabilities in the source code of the Minswap - AMM DEX V2 project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis 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:

- Testing the smart contracts against both common and uncommon attack vectors;
- 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.

DEPENDENCIES | MINSWAP - AMM DEX V2

Out Of Scope Dependencies

The `batcher` and `admin` roles within the protocol are granted significant authority. Each role requires possession of specific tokens to function effectively, their minting policies were not examined as part of this audit's scope.

Assumptions

We operate under the assumption that all mechanisms and protocols related to these roles and their associated tokens are correctly implemented and secure. This assumption extends to the belief that the minting policies, while not reviewed, are designed and function in a manner that supports the system's overall security and integrity.

Recommendations

We recommend all out-of-scope dependencies are carefully vetted to ensure they function as intended. Last, we recommend all assumptions about the behavior of the project are thoroughly reviewed and, if the assumptions do not match the intention of the protocol, documenting the intended behavior for review.

FINDINGS | MINSWAP - AMM DEX V2



10

Total Findings

0

Critical

3

Major

1

Medium

1

Minor

5

Informational

This report has been prepared to discover issues and vulnerabilities for Minswap - AMM DEX V2. Through this audit, we have uncovered 10 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
FAC-01	Creation Of Pools With Invalid Parameters	Logical Issue	Major	● Resolved
MIN-01	Logical Issue In Fee Settings	Incorrect Calculation, Logical Issue	Major	● Resolved
VAL-01	Centralization Related Risks	Centralization	Major	● Acknowledged
FAC-02	Pool Creation Allows Complete Asset Withdrawal	Design Issue	Medium	● Resolved
ORD-02	Missing Check On <code>io_ratio_denominator</code>	Volatile Code	Minor	● Resolved
AUT-01	Incorrect Comment	Coding Style	Informational	● Resolved
GLOBAL-01	Unit Test Documentation	Coding Style	Informational	● Acknowledged
MIN-02	<code>TODO</code> Comments	Coding Style	Informational	● Resolved
ORD-01	Missing Formulas For <code>WithdrawImbalance</code> And <code>PartialSwap</code>	Design Issue	Informational	● Resolved
ORD-03	Typos	Coding Style	Informational	● Resolved

FAC-01 | CREATION OF POOLS WITH INVALID PARAMETERS

Category	Severity	Location	Status
Logical Issue	● Major	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/validators/factory_validator.ak (Base): 158~160, 167~184, 220~224	● Resolved

Description

The validation process for creating pools currently focuses on matching the `PoolDatum` with the new pool's parameters. However, the validation checks for `total_liquidity`, `reserve_a`, and `reserve_b` are currently insufficient in preventing the establishment of pools with parameters that could obstruct future orders.

```

158     total_liquidity: pool_datum_total_liquidity,
159     reserve_a: pool_datum_reserve_a,
160     reserve_b: pool_datum_reserve_b,
```

The reserves of the pool are validated against `amount_a` and `amount_b`, with `total_liquidity` calculated as the square root of their product:

```

220     // Total Liquidity in PoolDatum must be sqrt(amount_a * amount_b)
221     pool_datum_total_liquidity == total_liquidity,
222     // Pool Reserve must be the same between datum and value
223     pool_datum_reserve_a == amount_a,
224     pool_datum_reserve_b == amount_b,
```

`amount_a` and `amount_b` are directly verified against the amounts in the pool output's value.

```

167     let estimated_amount_a =
168         value.quantity_of(
169             pool_output_value,
170             asset_a_policy_id,
171             asset_a_asset_name,
172         )
173     let amount_a =
174         if utils.is_ada_asset(asset_a_policy_id, asset_a_asset_name) {
175             estimated_amount_a - 3000000
176         } else {
177             estimated_amount_a
178         }
179     let amount_b =
180         value.quantity_of(
181             pool_output_value,
182             asset_b_policy_id,
183             asset_b_asset_name,
184         )
```

This validation process does not account for scenarios where assets provided in the redeemer are not included in the pool, potentially leading to `amount_a` being zero and `amount_b` being zero.

Scenario

If an asset specified in the redeemer is missing in the output, it results in a zero reserve for that asset and, consequently, zero `total_liquidity`.

This condition renders the pool dysfunctional, causing failures for all transaction types, notably making deposits unfeasible. Since the pool creation logic prevents duplicate pools, this flaw can be exploited to cause a service disruption, blocking the establishment of a functional pool for a specific asset pair.

Recommendation

To mitigate the risk of creating non-functional pools, we advise implementing additional validation checks to ensure that reserve values and total liquidity fall within practical ranges. Specifically, validations should preclude scenarios where reserves could be zero or negative and confirm that the computed liquidity supports viable pool operations. These measures will enhance the robustness of pool creation and ensure the protocol's integrity.

Alleviation

[CertiK, 2024/02/19]: The team heeded the advice and resolved the issue of reserves being set to zero or negative number, in commit [ec38cd102e6bdb6b8a2bfd489f850d0ffb97ec01](#):

MIN-01 | LOGICAL ISSUE IN FEE SETTINGS

Category	Severity	Location	Status
Incorrect Calculation, Logical Issue	● Major	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/ib/amm_dex_v2/pool_validation.ak (Base): 569~572, 585~588, 607~611, 624~628; minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/validators/factory_validator.ak (Base): 211~212; minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/validators/pool_validator.ak (Base): 113~114	● Resolved

Description

The function `pool_validation.validate_update_pool_datum_or_stake_credential()` is designed to ensure that updates to the pool datum comply with the specified requirements for fees. Specifically, it mandates that new fee values for `Profit Sharing` should fall within the range of 16.66% to 50%, and for `Trading Fee`, between 0.05% to 10%. Despite these checks, the validation functions `pool_validation.validate_trading_fee_percent()` and `pool_validation.validate_fee_sharing_percent()` currently do not safeguard against the possibility of setting both the numerator and denominator of fees to zero. This oversight can lead to critical failures within the protocol's fee computation mechanisms due to division by zero errors.

A similar issue is found in `factory_validator.validate_factory()`, as it also relies on `pool_validation.validate_trading_fee_percent()` to set fees when creating a new pool.

Recommendation

We recommend implementing additional validation within `pool_validation.validate_trading_fee_percent()` and `pool_validation.validate_fee_sharing_percent()` to ensure that neither the numerator nor the denominator of fee values can be set to zero. This precaution will help to eliminate the risk of rendering the protocol inoperative and ensure the integrity and functionality of fee computations.

Alleviation

[CertiK Team, 2024/02/14]: The team heeded the advice and resolved the issue in commit [242e522569305bd3fb5fac2d56079728f97faf6a](#).

VAL-01 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization	● Major	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/validators/factory_validator.ak (Base): 206~209; minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/validators/pool_validator.ak (Base): 25~26, 60~65, 102~103, 125~126	● Acknowledged

Description

In the validator `factory_validator.validate_factory()` the owner of an `admin` token can create pools with non-standard fees between 0.05% and 10%, contrary to regular users who can only choose 0.05%, 0.3%, or 1%.

In the validator `pool_validator.validate_pool()` the owner of an `admin` token can use the actions:

- `UpdatePoolFeeOrStakeCredential` to modify the pool fees or change the address of the pool;
- `WithdrawLiquidityShare` to withdraw protocol fees and send them to any address;

Any compromise to an `admin` account may allow a hacker to take advantage of this authority and :

- create pools with non-standard fees;
- modify important parameters from a pool;
- steal the protocol fees;

In the validator `pool_validator.validate_pool()` , the owner of a license token, i.e a `batcher` can use the actions:

- `Batching` to submit a batch of orders in a transaction;
- `MultiRouting` to trigger a multi swap order;

Any compromise to a `batcher` account may allow a hacker to take advantage of this authority and submit transactions, potentially allowing manipulation of the order of transactions.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged role and the token minting policy to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via decentralized mechanisms.

The team should ensure a total transparency about the `batcher` and `admin` roles, their mechanisms and the potential risk through articles or blog posts.

They should set clear expectations for how the batcher is supposed to behave (e.g. ruling out front-running), and clarify how it can be monitored to mitigate unexpected events.

■ Alleviation

[Minswap, 2024/02/23]: To mitigate the centralization risks, the following strategies are explored:

1. **License Expiration:** Assigning an expiration date to the license tokens ensures its permissions are temporary, preventing any entity from holding perpetual control.
2. **Multisig Minting:** The minting of licenses is managed through a 2/3 multisig script, utilizing Cardano's native multi-signature functionality.
3. **Future Transferability to Smart Contracts:** Licenses will be transferable to a smart contract in the future. Although a license owner currently holds significant permissions, transitioning ownership to a smart contract will subject those permissions to the contract's predefined rules, such as time-locks, decentralized governance, or Layer 2 consensus protocols.

[CertiK, 2024/03/01]: The license expiration mechanisms and the transferability to smart contracts have been enabled in commit: [47bdfd33df7d9c42777ad1a48126028c84373a72](#).

To prevent potential risk of cross-privilege issues between `admin` and `batcher` tokens, which arose from using timestamps as `TokenName` for both, the currency policy for the `admin` token has been updated in commit [9f1ac03dee95d9a3a4981f655ec5fc51987e87a9](#).

FAC-02 POOL CREATION ALLOWS COMPLETE ASSET WITHDRAWAL

Category	Severity	Location	Status
Design Issue	● Medium	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/validators/factory_validator.ak (Base): 186~187	● Resolved

Description

The pool creation mechanism precisely aligns the `pool_datum_total_liquidity` with the combined reserves of assets A and B, adhering strictly to the liquidity formulas. This equality allows liquidity providers to potentially fully withdraw their assets, leading to the depletion of the pool's reserves to zero. Due to existing safety measures that prevent operations with zero reserves, such a state would cause deposit and swap transactions to fail, making the pool unusable for further transactions.

Recommendation

To mitigate this issue, several strategies can be considered:

- **Facilitate Pool Replenishment:** Introducing mechanisms that allow for the replenishment of pool liquidity. This solution may require adjustments in the logic for deposit orders, potentially introducing complexity and associated risks.
- **Burn Minimal LP Tokens at Pool Creation:** A simpler and less intrusive solution involves burning a small number of LP tokens when the pool is created. This approach prevents the total withdrawal of liquidity and is a proven strategy for similar Automated Market Makers (AMMs).

Alleviation

[CertiK, 2024/02/27]: The team has implemented a mechanism that burns a small amount of LP tokens upon pool creation. This approach, which removes a minor quantity of LP tokens from the pool creator, is similar to the strategy utilized by UniswapV2, as detailed in their [whitepaper](#).

Commit: [dd9680488517bceb186ebc8d0ebd257c787e2b65](#).

ORD-02 | MISSING CHECK ON `io_ratio_denominator`

Category	Severity	Location	Status
Volatile Code	Minor	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/lib/amm_dex_v2/order_validation.ak (Base): 1159~1160	Resolved

Description

In `order_validation.apply_orders()` for the `PartialSwap` order, there is a redundancy in the validation check where `io_ratio_numerator` is evaluated twice:

```
1158         expect and {  
1159             io_ratio_numerator > 0,  
1160             io_ratio_numerator > 0,
```

This oversight means the check intended for `io_ratio_denominator` is mistakenly omitted, leading to a potential lapse in catching incorrect ratio values early in the validation process.

Recommendation

We recommend updating the validation logic to correctly check both `io_ratio_numerator` and `io_ratio_denominator` for positive values, as initially intended.

Alleviation

[CertiK, 2024/02/14]: The team heeded the advice and resolved the issue in commit [242e522569305bd3fb5fac2d56079728f97faf6a](#).

AUT-01 | INCORRECT COMMENT

Category	Severity	Location	Status
Coding Style	● Informational	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c3 7/validators/authen_minting_policy.ak (Base): 23	● Resolved

Description

The comment should be placed in front of redeemer `MintFactoryAuthen` instead of redeemer `CreatePool`.

Recommendation

We advise the team to perform related changes.

Alleviation

[CertiK, 2024/02/19]: The team heeded the advice and resolved the finding, commit: [ec38cd102e6bdb6b8a2bfd489f850d0ffb97ec01](#).

GLOBAL-01 | UNIT TEST DOCUMENTATION

Category	Severity	Location	Status
Coding Style	● Informational		● Acknowledged

Description

Code documentation can be improved by providing unit tests for the workflows. Unit tests could be used to cover and precisely test the program in its development evolution, to assert properties being held when code evolves. The current project only contain in-line tests for a few functions.

Documenting with unit tests allows precise expression of the expectations about the program. Examples may include expected cases, edge cases and ensure rejections of invalid data.

Recommendation

We recommend documenting the various program use cases with unit-tests and integration tests.

In addition, we did not see test results and coverages in the provided GitHub repository. It is good to have happy path tested but we strongly recommend to add more tests, including but not limited to, testing:

- error conditions,
- input corner cases,
- correct workflows

Alleviation

[CertiK, 2024/02/19]: The team acknowledged the finding but chose not to make changes.

MIN-02 | TODO COMMENTS

Category	Severity	Location	Status
Coding Style	● Informational	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/lib/amm_dex_v2/pool_validation.ak (Base): 415~416; minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/validators/aut hen_minting_policy.ak (Base): 64~65	● Resolved

Description

The codebase contains several `TODO` comments indicating unfinished tasks or features that require further development or revision. The presence of these comments makes it challenging to ascertain the completeness and current status of the implicated sections.

Recommendation

We recommend conducting a thorough review of all `TODO` comments in the code. For each comment, either complete the pending task or provide a detailed explanation in the documentation regarding its status and any planned actions. This approach will help clarify the code's current state and future development plans, ensuring a more polished and transparent project.

Alleviation

[CertiK, 2024/02/19]: The team heeded the advice and resolved the finding, commit: [ec38cd102e6bdb6b8a2bfd489f850d0ffb97ec01](#).

ORD-01 | MISSING FORMULAS FOR `WithdrawImbalance` AND `PartialSwap`

Category	Severity	Location	Status
Design Issue	● Informational	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/lib/amm_dex_v2/order_validation.ak (Base): 1240~1241	● Resolved

Description

The project documentation currently lacks detailed formulas for the `WithdrawImbalance` and `PartialSwap` orders. These sections are marked as "TODO", indicating that the information has yet to be provided. The absence of these formulas makes it difficult to verify that the features's implementations align with the expected behaviors and specifications. This gap in documentation requires attention to ensure clarity and completeness.

Recommendation

We recommend promptly providing the missing documentation for the `WithdrawImbalance` and `PartialSwap` features. This step is essential for us to verify that the implementations align with the intended specifications and ensure everything functions as expected.

Alleviation

[CertiK, 2024/02/20]: The formulas have been described in the documentation in commit [d51628e907f4c2a5b4f55f95c58a2a15bed066ef](#), however, the specification doc still contains multiple (TODO: Link formula section here).

[CertiK, 2024/02/20]: The team fully resolved the finding in commit [058d9f37657a30fae31895dd49f962e40af2a0ad](#).

ORD-03 | TYPOS

Category	Severity	Location	Status
Coding Style	● Informational	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c564588c37/ lib/amm_dex_v2/order_validation.ak (Base): 43, 78~79, 177~178, 270~271, 345~346, 349~350, 435~436, 628~629, 965~966, 992~993, 1026~1027	● Resolved

Description

In the file `order_validation.ak`, there's a recurring typographical error where the word `satisfied` is consistently misspelled as `sastified`. Typographical errors can impact the readability and understandability of the codebase and its documentation. Such inaccuracies could also pose challenges for future development. Specifically, they may complicate keyword searches, adversely affecting the maintainability and scalability of the code.

It's important to note that the instances identified may not represent all typos within the project. While small errors are inevitable, addressing them promptly ensures smoother future revisions and updates.

Recommendation

We recommend a thorough review and correction of typographical errors, beginning with the frequent misspelling identified in `order_validation.ak`, are recommended. Employing automated spell-check tools and peer review processes can aid in detecting and amending these mistakes.

Alleviation

[CertiK, 2024/02/19]: The team heeded the advice and resolved the finding, commit: [ec38cd102e6bdb6b8a2bfd489f850d0ffb97ec01](https://github.com/minswap/minswap/commit/ec38cd102e6bdb6b8a2bfd489f850d0ffb97ec01).

OPTIMIZATIONS | MINSWAP - AMM DEX V2

ID	Title	Category	Severity	Status
<u>MAT-01</u>	Potential Optimization In <div>math.calculate_withdraw_imbalance()</div>	Code Optimization	Optimization	<div> <div></div> Acknowledged </div>

MAT-01 | POTENTIAL OPTIMIZATION IN

`math.calculate_withdraw_imbalance()`

Category	Severity	Location	Status
Code Optimization	● Optimization	minswap-dex-v2-17ab42bcde1513a1138e3124906a22c56 4588c37/lib/amm_dex_v2/math.ak (Base): 216~217, 249~250, 282~283, 387~388, 413~414, 448~449, 483~484	● Acknowledged

Description

Within the `math.calculate_withdraw_imbalance()` function, the ratio of the amount of assets withdrawn is compared against an expected ratio. If they do not match, the necessary amount of assets to be swapped is calculated to align with the desired ratio. However, when the actual ratio of amounts is "close enough" to the expected ratio, the rounding caused by the use of integer operations (`Int`) in `math.calculate_withdraw_swap_amount()` may result in the computed adjustment amount being zero. Consequently, in situations where the ratios are not exactly equal but fall within a certain proximity, this scenario should be treated as if `ratio_a = ratio_b`, since the outcome will essentially be identical.

The similar process can also be optimized in function `calculate_deposit_amount()`.

Recommendation

We recommend identifying and defining a 'neighborhood' threshold that allows for bypassing unnecessary computations when the actual and expected ratios are sufficiently close. Addressing this optimization should be prioritized after the missing documentation, specifically, the detailed formulas and specifications, has been completed and provided. This approach ensures that any adjustments are made with a full understanding of the intended mathematical behavior.

Alleviation

[CertiK, 2024/02/16]: The team acknowledged the finding but chose not to make changes.

APPENDIX | MINSWAP - AMM DEX V2

Finding Categories

Categories	Description
Coding Style	Coding Style findings may not affect code behavior, but indicate areas where coding practices can be improved to make the code more understandable and maintainable.
Incorrect Calculation	Incorrect Calculation findings are about issues in numeric computation such as rounding errors, overflows, out-of-bounds and any computation that is not intended.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.
Design Issue	Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories.

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 | CERTIK

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.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR

UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

CertiK | Securing the Web3 World

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

