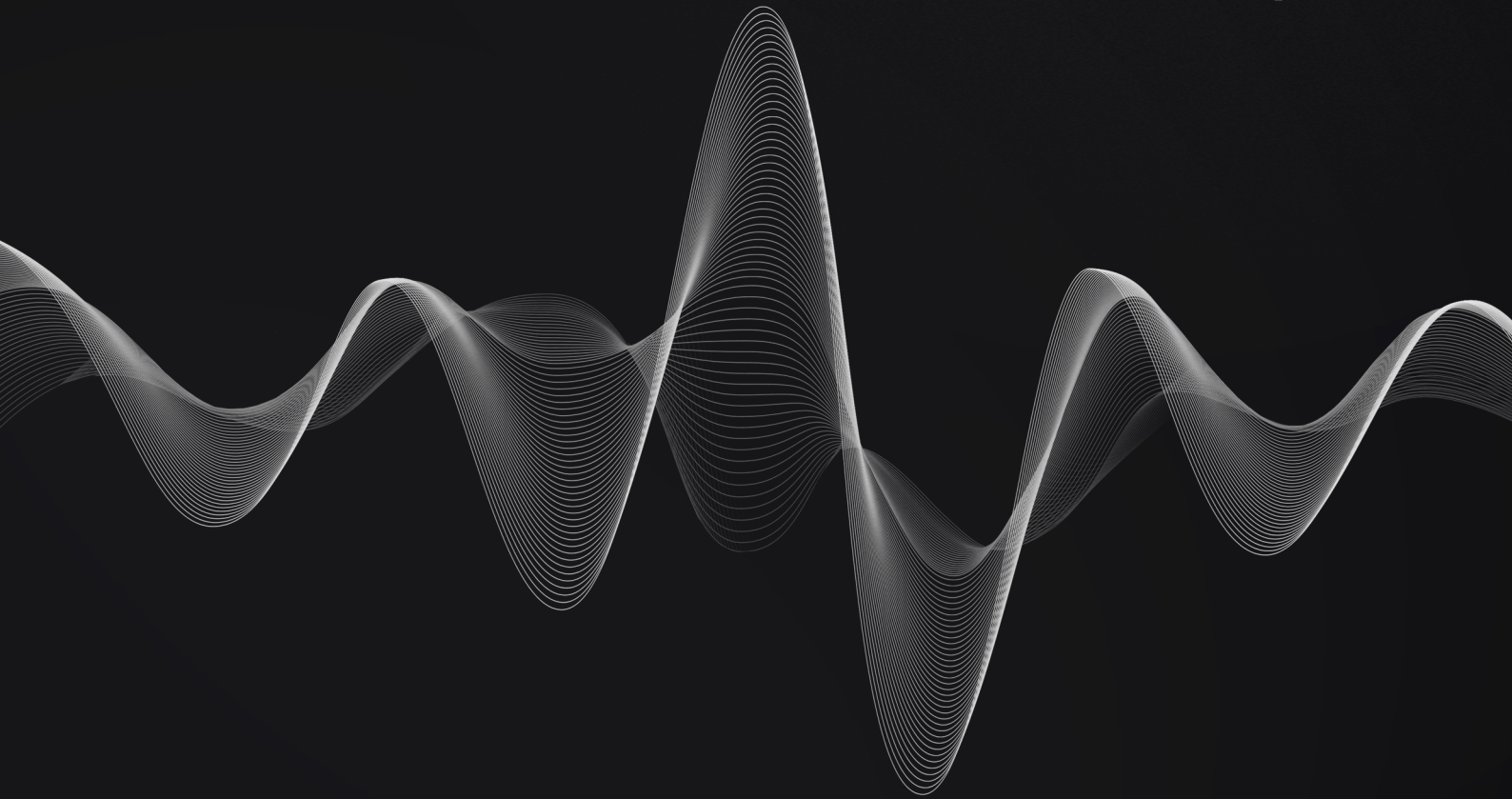


Primex Finance

Primex Protocol Bug Review Audit Report




Document Control

PUBLIC

FINAL(v2.1)

Audit_Report_PRMX-BUG_FINAL_21

Sep 15, 2024		v0.1	Michał Bazyli: Initial draft
Sep 15, 2024		v0.2	Michał Bazyli: Added findings
Sep 16, 2024		v1.0	Charles Dray: Approved
Sep 16, 2024		v1.1	Michał Bazyli: Reviewed findings
Oct 11, 2024		v1.2	João Simões: Reviewed contract updates
Oct 14, 2024		v2.0	Charles Dray: Finalized
Oct 14, 2024		v2.1	Charles Dray: Published

Points of Contact	Oleksandr Marukhnenko	Primex Finance	alex@primex.finance
	Charles Dray	Resonance	charles@resonance.security
Testing Team	Michał Bazyli	Resonance	michal@resonance.security
	João Simões	Resonance	joao@resonance.security
	Ilan Abitbol	Resonance	ilan@resonance.security

Copyright and Disclaimer

© 2024 Resonance Security, Inc. All rights reserved.

The information in this report is considered confidential and proprietary by Resonance and is licensed to the recipient solely under the terms of the project statement of work. Reproduction or distribution, in whole or in part, is strictly prohibited without the express written permission of Resonance.

All activities performed by Resonance in connection with this project were carried out in accordance with the project statement of work and agreed-upon project plan. It's important to note that security assessments are time-limited and may depend on information provided by the client, its affiliates, or partners. As such, the findings documented in this report should not be considered a comprehensive list of all security issues, flaws, or defects in the target system or codebase.

Furthermore, it is hereby assumed that all of the risks in electing not to remedy the security issues identified henceforth are sole responsibility of the respective client. The acknowledgement and understanding of the risks which may arise due to failure to remedy the described security issues, waives and releases any claims against Resonance, now known or hereafter known, on account of damage or financial loss.

Contents

1 Document Control	2
Copyright and Disclaimer	2
2 Executive Summary	4
System Overview	4
Repository Coverage and Quality.....	4
3 Target	5
4 Methodology	6
Severity Rating.....	7
Repository Coverage and Quality Rating.....	8
5 Findings	9
Gas Price Not Set For Arbitrum.....	11
minPositionSizeMultiplier Misinterpretation.....	12
Lack Of Event Emission In TraderBalanceVault.....	13
Missing extraParams In OpenPositionData.....	14
Incorrect Fee In Health Calculation After Upgrade	15
Incorrect Calculations For minFeeRestrictions	16
Transaction Fails Due To Price Difference	17
Missing Estimation Of minProtocolFee For Liquidation	18
Updated Admin Roles.....	19
Missing protocolFeeCoefficient In minPositionSize Calculation.....	20
Missing Revert Case For Uniswap v3 TWAP Oracle	21
Missing Reinitialization On SwapManager And BatchManager Due To Upgrade	22
Limit Orders Work Incorrectly Due To Fee Structure.....	23
Overly Restrictive Requirement Leads To Failed Token Swaps	24
Impossible To Create Orders For Assets With Pull Oracles	25
Parameter Name Mismatch In setPoolUpdateInterval.....	26
Incorrect Token Amount Used In Protocol Fee Calculation For Batch Closing.....	27
Uniform Health Check During Liquidation	28
Impossible To Execute Limit Orders Through PrimexUpkeep	29
A Proof of Concepts	30

Executive Summary

Primex Finance contracted the services of Resonance to conduct a comprehensive security audit of their smart contracts between September 9, 2024 and September 16, 2024. The primary objective of the assessment was to verify applied fixes and identify any potential security vulnerabilities and ensure the correct functioning of smart contract operations.

During the engagement, Resonance allocated 2 engineers to perform the security review. The engineers, including an accomplished professional with extensive proficiency in blockchain and smart-contract security, encompassing specialized skills in advanced penetration testing, and in-depth knowledge of multiple blockchain protocols, devoted 5 days to the project. The project's test targets, overview, and coverage details are available throughout the next sections of the report.

The ultimate goal of the audit was to provide Primex Finance with a detailed summary of the findings, including any identified vulnerabilities, and recommendations to mitigate any discovered risks. The results of the audit are presented in detail further below.



System Overview

Primex Finance is a decentralized exchange platform powered by smart contract where users can lend and swap tokens, perform trading operations, and earn rewards by utilizing the protocol.

The system is composed of several components: the bucket infrastructure where all the lending and borrowing happens; the trading managers that implement logic to swap, open and close positions, with or without conditions; and the protocol management where Primex admin control emergency pauses, propositions and executions.

As a general workflow, a lender is able to stake collateral on a bucket and receive yield-bearing tokens. The collateral can be used by traders to borrow and open margin positions. The keepers handle the conditions that occur over time, such as, opening limit orders when price as reached the limit price, liquidating users with underwater positions, closing trades with set stop-losses, etc.

By using the different components of the protocol, all actors receive rewards that incentivise further usage. One of such rewards is the PMX token that offers utility and governance capabilities on the protocol.

Additional features were implemented such as, the integration of Enso and arbitrary contract swappers, the integration of the Optimistic chain and Supra oracle, and flash loan capabilities.



Repository Coverage and Quality

Code

N/A

Tests

N/A

Documentation

N/A

Target

The objective of this project is to conduct a comprehensive review and security analysis of the smart contracts that are contained within the specified repository.

The following items are included as targets of the security assessment:

- Repository (main): [primex-finance/primex-protocol/src/contracts](#)
- Hash: e95e2b665b6c6ed65d628227ae4683fe85ded7e3
- Repository: [primex-finance/primex_contracts/src/contracts](#)
- Hash: a4afb1763211c8ab2f19a127adf68cf35ffa5eb

The following items are excluded:

- External and standard libraries
- Files pertaining to the deployment process
- Financial-related attack vectors

Methodology

In the context of security audits, Resonance's primary objective is to portray the workflow of a real-world cyber attack against an entity or organization, and document in a report the findings, vulnerabilities, and techniques used by malicious actors. While several approaches can be taken into consideration during the assessment, Resonance's core value comes from the ability to correlate automated and manual analysis of system components and reach a comprehensive understanding and awareness with the customer on security-related issues.

Resonance implements several and extensive verifications based off industry's standards, such as, identification and exploitation of security vulnerabilities both public and proprietary, static and dynamic testing of relevant workflows, adherence and knowledge of security best practices, assurance of system specifications and requirements, and more. Resonance's approach is therefore consistent, credible and essential, for customers to maintain a low degree of risk exposure.

Ultimately, product owners are able to analyze the audit from the perspective of a malicious actor and distinguish where, how, and why security gaps exist in their assets, and mitigate them in a timely fashion.

Source Code Review - Solidity EVM

During source code reviews for Web3 assets, Resonance includes a specific methodology that better attempts to effectively test the system in check:

1. Review specifications, documentation, and functionalities
2. Assert functionalities work as intended and specified
3. Deploy system in test environment and execute deployment processes and tests
4. Perform automated code review with public and proprietary tools
5. Perform manual code review with several experienced engineers
6. Attempt to discover and exploit security-related findings
7. Examine code quality and adherence to development and security best practices
8. Specify concise recommendations and action items
9. Revise mitigating efforts and validate the security of the system

Additionally and specifically for Solidity EVM audits, the following attack scenarios and tests are recreated by Resonance to guarantee the most thorough coverage of the codebase:

- Reentrancy attacks
- Frontrunning attacks
- Unsafe external calls
- Unsafe third party integrations
- Denial of service
- Access control issues

- Inaccurate business logic implementations
- Incorrect gas usage
- Arithmetic issues
- Unsafe callbacks
- Timestamp dependence
- Mishandled panics, errors and exceptions



Severity Rating

Security findings identified by Resonance are rated based on a Severity Rating which is, in turn, calculated off the **impact** and **likelihood** of a related security incident taking place. This rating provides a way to capture the principal characteristics of a finding in these two categories and produce a score reflecting its severity. The score can then be translated into a qualitative representation to help customers properly assess and prioritize their vulnerability management processes.

The **impact** of a finding can be categorized in the following levels:

1. Weak - Inconsequential or minimal damage or loss
2. Medium - Temporary or partial damage or loss
3. Strong - Significant or unrecoverable damage or loss

The **likelihood** of a finding can be categorized in the following levels:

1. Unlikely - Requires substantial knowledge or effort or uncontrollable conditions
2. Likely - Requires technical knowledge or no special conditions
3. Very Likely - Requires trivial knowledge or effort or no conditions

		Likelihood		
		Very Likely	Likely	Unlikely
Impact	Strong	Critical	High	Medium
	Medium	High	Medium	Low
	Weak	Medium	Low	Info



Repository Coverage and Quality Rating

The assessment of Code, Tests, and Documentation coverage and quality is one of many goals of Resonance to maintain a high-level of accountability and excellence in building the Web3 industry. In Resonance it is believed to be paramount that builders start off with a good supporting base, not only development-wise, but also with the different security aspects in mind. A product, well thought out and built right from the start, is inherently a more secure product, and has the potential to be a game-changer for Web3's new generation of blockchains, smart contracts, and dApps.

Accordingly, Resonance implements the evaluation of the code, the tests, and the documentation on a score **from 1 to 10** (1 being the lowest and 10 being the highest) to assess their quality and coverage. In more detail:

- Code should follow development best practices, including usage of known patterns, standard libraries, and language guides. It should be easily readable throughout its structure, completed with relevant comments, and make use of the latest stable version components, which most of the times are naturally more secure.
- Tests should always be included to assess both technical and functional requirements of the system. Unit testing alone does not provide sufficient knowledge about the correct functioning of the code. Integration tests are often where most security issues are found, and should always be included. Furthermore, the tests should cover the entirety of the codebase, making sure no line of code is left unchecked.
- Documentation should provide sufficient knowledge for the users of the system. It is useful for developers and power-users to understand the technical and specification details behind each section of the code, as well as, regular users who need to discern the different functional workflows to interact with the system.

Findings

During the security audit, several findings were identified to possess a certain degree of security-related weaknesses. These findings, represented by unique IDs, are detailed in this section with relevant information including Severity, Category, Status, Code Section, Description, and Recommendation. Further extensive information may be included in corresponding appendices should it be required.

An overview of all the identified findings is outlined in the table below, where they are sorted by Severity and include a **Remediation Priority** metric asserted by Resonance's Testing Team. This metric characterizes findings as follows:

- ||||| **"Quick Win"** Requires little work for a high impact on risk reduction.
- |||| **"Standard Fix"** Requires an average amount of work to fully reduce the risk.
- ||| **"Heavy Project"** Requires extensive work for a low impact on risk reduction.

Findings ID	Description	Severity	Status
RES-01	Gas Price Not Set For Arbitrum		Resolved
RES-02	minPositionSizeMultiplier Misinterpretation		Resolved
RES-03	Lack Of Event Emission In TraderBalanceVault		Resolved
RES-04	Missing extraParams In OpenPositionData		Resolved
RES-05	Incorrect Fee In Health Calculation After Upgrade		Resolved
RES-06	Incorrect Calculations For minFeeRestrictions		Resolved
RES-07	Transaction Fails Due To Price Difference		Resolved
RES-08	Missing Estimation Of minProtocolFee For Liquidation		Resolved
RES-09	Updated Admin Roles		Resolved
RES-10	Missing protocolFeeCoefficient In minPositionSize Calculation		Resolved
RES-11	Missing Revert Case For Uniswap v3 TWAP Oracle		Resolved
RES-12	Missing Reinitialization On SwapManager And BatchManager Due To Upgrade		Resolved
RES-13	Limit Orders Work Incorrectly Due To Fee Structure		Resolved
RES-14	Overly Restrictive Requirement Leads To Failed Token Swaps		Resolved

RES-15	Impossible To Create Orders For Assets With Pull Oracles		Resolved
RES-16	Parameter Name Mismatch In setPoolUpdateInterval		Resolved
RES-17	Incorrect Token Amount Used In Protocol Fee Calculation For Batch Closing		Resolved
RES-18	Uniform Health Check During Liquidation		Resolved
RES-19	Impossible To Execute Limit Orders Through PrimexUpkeep		Resolved



Gas Price Not Set For Arbitrum

Info

RES-PRMX-BUG01

Gas Optimization

Resolved

Code Section

[PRIM-6692: fix gas price calculation on Arbitrum](#)

Description

Issue identified by Primex:

Gas price oracle was not set for Arbitrum, and `defaultMaxGasPrice` was used instead of it that caused incorrect min protocol fee and min position size calculations. We've added a wrapper contract on top of Arb gas price precompiles that returns gas price in the required format.

Recommendation

Not specified.

Status

The issue has been fixed in `2e85703cb69a5c775c98b9b2f9fc980579a6a317`.



minPositionSizeMultiplier Misinterpretation

Info

RES-PRMX-BUG02

Code Quality

Resolved

Code Section

- [PRIM-6688: remove requires for positionSizeMultiplier](#)
- [fix\(PRIM-0000\): add oz comment](#)

Description

Issue identified by Primex:

We renamed `minPositionSizeMultiplier` to `minPositionSizeAddend` because the old name didn't correspond to its meaning, and irrelevant restrictions were removed.

Recommendation

Not specified.

Status

The issue has been fixed in `5c23d513de4944c8e3d5624865a3bc30b6392fbb`.



Lack Of Event Emission In TraderBalanceVault

Info

RES-PRMX-BUG03

Code Quality

Resolved

Code Section

- [PRIM-6724: add events to the vault](#)

Description

Issue identified by Primex:

The following functions do not emit relevant events after executing sensitive actions:

- `withdrawFrom`
- `increaseLockedBalance`
- `topUpAvailableBalance`

Recommendation

Not specified.

Status

The issue has been fixed in 7d799ab624637a425e2a51748032c49d4f75141c.



Missing extraParams In OpenPositionData

Info

RES-PRMX-BUG04

Code Quality

Resolved

Code Section

- PRIM: add extraParams to OpenPositionData struct in PrimexLens

Description

Issue identified by Primex:

The OpenPositionData struct had no extraParams field in `src/contracts/interfaces/IPrimexLens.sol`

Recommendation

Not specified.

Status

The issue has been fixed in 719a9adb7ef89470b84fef3f4fb792266b3d0d05.



Incorrect Fee In Health Calculation After Upgrade

Info

RES-PRMX-BUG05

Data Validation

Resolved

Code Section

- [PRIM-6712: fix health calc in the batch manager](#)

Description

Issue identified by Primex:

The positions opened before the upgrade paid the protocol fee at the moment of opening, and the fee from them is not taken again upon closing, but this fee was taken into account in the health calculation. A check was added to avoid it.

Recommendation

Not specified.

Status

The issue has been fixed in 43df7608e8705794e012c9ef09a3c318832b5369.



Incorrect Calculations For minFeeRestrictions

Info

RES-PRMX-BUG06

Business Logic

Resolved

Code Section

- PRIM: [update getL1BaseLengthForTradingOrderType](#)

Description

Issue identified by Primex:

Method `getL1BaseLengthForTradingOrderType` returned the sum of lengths for position opening and closing for limit orders that caused incorrect calculations of `minProtocolFee` for these actions separately.

Recommendation

Not specified.

Status

The issue has been fixed in [818ae7f4fb24958a335c1f793264a1c38b66c27a](#).



Transaction Fails Due To Price Difference

Info

RES-PRMX-BUG07

Data Validation

Resolved

Code Section

- [PRIM-6654: leverage tolerance](#)
- [PRIM- 6654: fixes](#)

Description

Issue identified by Primex:

The borrowed amount for margin limit orders was calculated during the order execution based on the oracle prices. At the same time, keepers calculate the swap route before the transaction execution, and if the `amountIn` used in the route and the amount calculated in tx differs, the transaction fails. To fix it, we allowed keepers to set the borrowed amount, but we also check that the leverage of the opened position can't differ from the leverage of the order for more than `leverageTolerance` percentage.

Recommendation

Not specified.

Status

The issue has been fixed in `5b6bc26a3df80923fbcee652d9a8aa16fdf23788`.



Missing Estimation Of minProtocolFee For Liquidation

Info

RES-PRMX-BUG08

Arithmetic Issues

Resolved

Code Section

- [PRIM: add contract primexLensPart2](#)

Description

Issue identified by Primex:

Added a new method to Lens to calculate min protocol fee for liquidation.

Recommendation

Not specified.

Status

The issue has been fixed in 5e65a6fab66eb5f40082e0ed1cd4573132c3f6f2.



Updated Admin Roles

Info

RES-PRMX-BUG09

Access Control

Resolved

Code Section

- [PRIM-6596: change role admins](#)
- [PRIM-6652: change admins for interest rate strategy](#)
- [Fix/prim 6407 fix admin roles](#)

Description

Issue identified by Primex:

Change admin roles for some methods.

Recommendation

Not specified.

Status

The issue has been fixed in 5e33b9d59d13250c49594cfb2fe4adb86e0bfce7.



Missing protocolFeeCoefficient In minPositionSize Calculation

Info

RES-PRMX-BUG10

Data Validation

Resolved

Code Section

- [PRIM-6626: add protocolFeeCoefficient to minPositionSize](#)

Description

Issue identified by Primex:

The protocolFeeCoefficient was missed in minPositionSize calculation, but this coefficient is a part of min protocol fee, which defines the min position size.

Recommendation

Not specified.

Status

The issue has been fixed in e15a23e9e3cf890956bb71f8b43ec73a040a1033.



Missing Revert Case For Uniswap v3 TWAP Oracle

Info

RES-PRMX-BUG11

Data Validation

Resolved

Code Section

- [fix twap contract](#)

Description

Issue identified by Primex:

Add revert to Uni v3 TWAP oracle for the case if the pool is unavailable.

Recommendation

Not specified.

Status

The issue has been fixed in 0c68d50d5e5b4104e0283e36461c9546120a7a48.



Missing Reinitialization On SwapManager And BatchManager Due To Upgrade

Info

RES-PRMX-BUG12

Data Validation

Resolved

Code Section

- [fix\(PRIM-6713\): add reinitilize functions](#)

Description

Issue identified by Primex:

The `initialize` functions of `SwapManager` and `BatchManager` check the interfaces of the contracts they receive as parameters, including `IPositionManagerV2` and `IPriceOracleV2`, which will be added with this upgrade. However, the `initialize` method was called before the upgrade of the `PositionManager` and `PriceOracle` contracts, causing the check to fail. To resolve this, the initialization was split into two steps, with the second step executed after the upgrade of these contracts.

Recommendation

Not specified.

Status

The issue has been fixed in `f5fbb5f23d1d8234dc69e9e6208212801cd0f3b5`.



Limit Orders Work Incorrectly Due To Fee Structure

Info

RES-PRMX-BUG13

Data Validation

Resolved

Code Section

- [PRIM-6703: fix problems with old orders after upgrade](#)

Description

Issue identified by Primex:

After the changes in fee structure, limit orders created before the upgrade worked incorrectly - the fee associated with them was locked after the order execution or editing, also the order became unexecutable and unclosable after editing.

Recommendation

Not specified.

Status

The issue has been fixed in 1dc27d3f1b256be4414ec7186e11c2b1756affd4.



Overly Restrictive Requirement Leads To Failed Token Swaps

Info

RES-PRMX-BUG14

Data Validation

Resolved

Code Section

- [PRIM: fix swapWithArbitraryDex](#)

Description

Issue identified by Primex:

It was impossible to swap tokens through an arbitrary contract using `DexAdapter.swapExactTokensForTokens` as it required that the router had to be whitelisted. This restriction was removed.

Recommendation

Not specified.

Status

The issue has been fixed in `0e762b9071f114ce8fd8ee6426e52decae070f07`.



Impossible To Create Orders For Assets With Pull Oracles

Info

RES-PRMX-BUG15

Data Validation

Resolved

Code Section

- [PRIM-6628: Incorrect arguments in createLimitOrder](#)
- [PRIM-6637: add pull oracle data to getPositionMaxDecrease](#)

Description

Issue identified by Primex:

In `createLimitOrder`, there is a check that the expected position size is higher than `minPositionSize`. However this method dont include `pullOracleData` so it was impossible to create an order for assets with pull oracles.

Recommendation

Not specified.

Status

The issue has been fixed in 1050bb46379c1ab8cae1fd324eb2d6396bf10216.



Parameter Name Mismatch In setPoolUpdateInterval

Info

RES-PRMX-BUG16

Code Quality

Resolved

Code Section

- [PRIM: fix misprint in parameter](#)

Description

Issue identified by Primex:

The `setPoolUpdateInterval` function in `src/contracts/UniswapPriceFeed/UniswapPriceFeed.sol` is designed to set `poolUpdateInterval`. However it will not work due to parameter name mismatch.

Recommendation

Not specified.

Status

The issue has been fixed in `8f25466c97adf9483fea737f43b1897a94d53b5e`.



Incorrect Token Amount Used In Protocol Fee Calculation For Batch Closing

Info

RES-PRMX-BUG17

Data Validation

Resolved

Code Section

- [PRIM-6688: remove requires for positionSizeMultiplier](#)

Description

Issue identified by Primex:

An incorrect token amount was used in the protocol fee calculation for batch closing, `tokenOut` and `amountIn` were used instead of `tokenOut` and `amoutOut`.

Recommendation

Not specified.

Status

The issue has been fixed in `f0ec9af778ab530f9134e0f91fbd56607f583125`.



Uniform Health Check During Liquidation

Info

RES-PRMX-BUG18

Business Logic

Resolved

Code Section

- [PRIM-6586: uniform health check during liquidation](#)

Description

Issue identified by Primex:

There is a difference in `minProtocolFee` calculation for single and batch position closing which leads to different health values and the liquidation price for the same position depending on the closing method.

Recommendation

Not specified.

Status

The issue has been fixed in 5985db454fe0cbc146ee5c01e3fdb6cb5fa258e.



Impossible To Execute Limit Orders Through PrimexUpkeep

Info

RES-PRMX-BUG19

Code Quality

Resolved

Code Section

- [fix primex upkeep](#)

Description

Issue identified by Primex:

Executing a limit order through primexUpkeep is not possible as the method is not payable, and it cannot pay the Pyth pull oracle fee in the native token.

Recommendation

Not specified.

Status

The issue has been fixed in `ff1ba0718c8a2db0baf0abbd4a869e76d6333463`.

Proof of Concepts

No Proof-of-Concept was deemed relevant to describe findings in this engagement.