Pump Science Solana Program Update Review and Security Analysis

Client: Pump Science

Date: 9th March 2025

Version: 1.0

Table of Contents

Security Review Information

Repository	pump-science-contract
From commit	f1916a44e031513f153ee4df60ac75727e47b7e9
To commit	93d25939afad3163d327aea3f21a5f3a9bd064b9
Remediation	6368031f640ce77c67b587f9afa7758928955a6d
Scope	pump-science-contract/programs/pump-science/*
Version	Final Report (v.1.0)
Date	9th March 2025

Executive summary

As of March 9, 2025, our comprehensive security review of the Pump Science protocol has been concluded. The assessment identified **1** low severity issue, which was acknowledged by the Pump Science team.

Introduction

Torii Security has been commissioned by Pump Science to conduct a Solana program update security review, focusing on the robustness, security, and efficiency of the program's implementation. This assessment specifically audited the new features introduced between commit *f1916a44e031513f153ee4df60ac75727e47b7e9* and commit *93d25939afad3163d327aea3f21a5f3a9bd064b9*, following the previous audit. The review aims to evaluate these newly added features and has the following goals:

- **Verify Protocol Integrity**: Assess the program's operation against its design specifications to ensure it functions correctly and efficiently within the Solana ecosystem. This includes evaluating its interaction with other protocols and services on the Solana network.
- **Identify Security Vulnerabilities**: Uncover potential security weaknesses that could be exploited by malicious actors, including but not limited to, flaws in program logic, transaction handling, and external dependencies.
- **Detect Program Bugs**: Identify bugs and glitches in the code that may result in unintended or erratic program behavior, potentially compromising its performance or security.
- **Provide Improvement Recommendations**: Offer actionable advice to enhance the program's security posture, efficiency, and code clarity, aiming to fortify it against current and future security threats while improving maintainability and scalability.

Exclusions from This Review

While the security review conducted by Torii Security on behalf of Pump Science provides a comprehensive analysis of the Solana Program's architecture, codebase, and security posture, certain aspects are beyond the scope of this audit. These exclusions are critical for stakeholders to understand, as they may require separate consideration and evaluation. The following areas have not been verified as part of this security review:

- **Deployment and Program Upgrade Process:** The procedures and mechanisms for deploying the Solana Program to the network and subsequent upgrades or modifications to the program are not covered. This includes the validation of deployment scripts, migration strategies, and the security of upgradeable contract mechanisms.
- **Keys Management**: The management, storage, and security practices for keys, including administrator keys and those used for program interactions, are outside the scope of this review. This encompasses both the technical and procedural safeguards in place to protect keys from unauthorized access or misuse.
- **Previous Features Audited**: This review did not reassess features that were already audited in prior security reviews conducted by other firms. The focus of this audit was strictly on the new modifications introduced in the specified commits, ensuring that they align with security best practices and do not introduce vulnerabilities.

Protocol Overview

Pump Science is a crypto-powered research protocol designed to maximize human healthspan rather than just lifespan. The protocol leverages tokenized interventions and market-driven research funding to advance scientific discoveries while integrating decentralized finance mechanics.

Key Features Introduced in the Latest Update

This security review specifically focuses on new features introduced between commit *f1916a44e031513f153ee4df60ac75727e47b7e9* and commit *93d25939afad3163d327aea3f21a5f3a9bd064b9*. The latest changes enhance fee structures, governance roles, and pool mechanisms to improve the protocol's flexibility and sustainability.

New Global Configuration Fields
Three new fields were added to the global_config PDA:

- *token_allo_receiver* – A dedicated wallet to receive 50M tokens allocated during migration (previously assigned to fee_receiver).
- *meteora_fee_authority* – A wallet designated as the authority to claim fees from Meteora (previously controlled by fee_receiver).
- *dev_fee_allo_bps* – Determines the fee share allocation of Meteora LP fees for the creator of the bonding curve.

These modifications ensure a clearer separation of roles for governance and fee distribution.

Migration Mechanism
A new *migrate_global* instruction was introduced to enable seamless PDA migration.
This instruction ensures existing configurations transition smoothly without requiring protocol resets.

Updates to create_pool and lock_pool Instructions
Modifications to pool creation and locking mechanisms allow the new global configuration fields to be properly integrated and enforced.

Adjusted Fee Structures

- The developer buy fee was increased from 0% to 1%.
- The fee reduction mechanism was extended:
    - Before: Linear fee reduction from slots 150 to 250.
    - Now: Extended reduction from slots 1000 to 2000, creating a more gradual and sustained transition.

Methodology

Issue Severity Classification

This report differentiates identified issues into distinct severity levels, each reflecting the potential impact on the system's security and overall functionality.

Severity	Description
Critical	Issues that present an immediate and severe threat, such as significant financial loss, irreversible locking of funds, or catastrophic system failure. These vulnerabilities require urgent remediation.
High	Bugs or vulnerabilities that could disrupt the correct operation of the system, potentially leading to incorrect states or temporary denial of service. Prompt attention and corrective action are necessary.
Medium	Issues that indicate deviations from best practices or suboptimal use of system primitives. While they may not pose immediate security threats, these issues could lead to vulnerabilities or inefficiencies if unaddressed.
Low	Minor concerns that have anegligible impact on system security or functionality. These may include inefficiencies or minor deviations from best practices that are unlikely to affect the system's operation significantly.
Informational	Suggestions related to design decisions, potential enhancements, or optimizations that do not have adirect impact on security. Implementing these recommendations may improve aspects such as usability or code readability but is not essential for system security.

Issue Status Definitions

Each issue is assigned astatus reflecting its current resolution stage.

Status	Description
Pending	The issue has been identified but not yet reviewed or addressed by the development team.
Acknowledged	The development team has recognized the issue but has not completed its resolution.
Resolved	The issue has been fully addressed, with implemented changes verified for effectiveness.

Findings Summary

ID	Title	Severity	Status
TS-L1	Developer Meteora Fee Allocation Affects Existing Pools	Low	Acknowledged
TS-I1	Hardcoded Developer Buy Fee	Informational	Acknowledged
TS-I2	Deprecated Code Comments	Informational	Resolved
TS-I3	Outdated Comment	Informational	Resolved
TS-I4	Missing Validation For New Global Parameters	Informational	Resolved
TS-I5	Incomplete Event Logging	Informational	Resolved

Detailed Findings

TS-L1 - Developer Meteora Fee Allocation Affects Existing Pools

Classification

Severity: Low
Status: Acknowledged

Description

The *dev_fee_allo_bps* parameter, which determines the allocation of the Meteora LP fee share for the bonding curve creator, is currently stored in *global_config*. This could lead to unintended consequences where changes to this parameter affect previously created pools rather than applying only to new ones.

Impact Scenario

1. A creator sets dev_fee_allo_bps to 2% and launches a pool.
2. Users invest, expecting this allocation to remain unchanged.
3. The creator later reduces dev_fee_allo_bps to 1% via global configuration updates.
4. This change retroactively affects all existing pools, potentially leading to unexpected outcomes for investors.

Recommendation

Store *dev_fee_allo_bps* in the *BondingCurve* struct instead of *global_config*, ensuring that changes apply only to newly created pools and do not affect previously deployed ones.

TS-I1 - Hardcoded Developer Buy Fee

Classification

Severity: Informational
Status: Acknowledged

Description

The developer buy fee is currently hardcoded at 1%. This limits flexibility and requires a contract upgrade if future changes are needed.

programs/pump-science/src/instructions/curve/swap.rs

```
msg!("Dev buy");
fee_bps = 100; // 1% = 100 basis points
fee_lamports = exact_in_amount
   .checked_mul(fee_bps)
   .unwrap()
   .checked_div(10000)
   .unwrap();
buy_amount_applied = exact_in_amount - fee_lamports;
```

Recommendation

Make the developer buy fee a configurable parameter instead of a hardcoded value.

TS-I2 - Deprecated Code Comments

Classification

Severity: Informational
Status: Resolved

Description

Old, commented-out code remains in the contract, which can lead to confusion and reduce code maintainability.

Affected code:
programs/pump-science/src/instructions/migration/create_pool.rs

```
// #[account(
//     mut,
//     associated_token::mint = token_b_mint,
//     associated_token::authority = fee_receiver
// )]
// pub fee_receiver_token_account: Box<Account<'info, TokenAccount>>,
#[account(
    mut,
    associated_token::mint = token_b_mint,
    associated_token::authority = global.token_allo_receiver
)]
pub token_allo_receiver_token_account: Box<Account<'info, TokenAccount>>,
```

Recommendation

Remove outdated commented-out code to improve code clarity and maintainability.

Remediation

The issue was fixed in the *121b900f2b8e310bc4b3668f86d879c80c2fe00c* commit.

TS-I3 - Outdated Comment

Classification

Severity: **Informational**
Status: **Resolved**

Description

A comment in *curve.rs* incorrectly refers to old slot values, which could lead to misunderstandings when reviewing the fee logic.

programs/pump-science/src/state/bonding_curve/curve.rs

```
// Calculate the minimum fee bps (at slot 250) scaled by 1_000_000 for
precision
```

Recommendation

Update the comment to reflect the correct slot range.

Remediation

The issue was fixed in the *121b900f2b8e310bc4b3668f86d879c80c2fe00c* commit.

TS-I4 - Missing Validation For New Global Parameters

Classification

Severity: **Informational**
Status: **Resolved**

Description

The *validate_settings* function (*programs/pump-science/src/state/global.rs*) does not validate newly added fields, potentially allowing misconfigured values to be set when updating *GlobalConfig* with *set_params* instruction.

Recommendation

Update *validate_settings* to include validation for the new fields introduced in the *global_config* update.

Remediation

The issue was fixed in the *121b900f2b8e310bc4b3668f86d879c80c2fe00c* commit.

TS-I5 - Incomplete Event Logging

Classification

Severity: **Informational**
Status: **Resolved**

Description

The event emitted during *GlobalConfig* updates (*set_params* instruction) does not include the newly added parameters, which reduces transparency and debugging capability.

Recommendation

Modify the event structure in the *into_event* function to include all new parameters for improved logging and visibility.

Remediation

The issue was fixed in the *121b900f2b8e310bc4b3668f86d879c80c2fe00c* commit.

Disclaimer

This report has been prepared in accordance with the current best practices and standards applicable at the time of its preparation. It is intended to provide an analysis and evaluation of the subject matter based on the information available, including any potential vulnerabilities, issues, or risks identified during the assessment process. The scope of this analysis is limited to the data, documents, and materials provided for review, and the conclusions drawn are based on the status of the information at the time of the report.

No representation or warranty, express or implied, is made as to the absolute completeness or accuracy of the information contained in this report. It is important to acknowledge that the findings, conclusions, and recommendations presented are subject to change should there be any modifications to the subject matter. This report should not be viewed as aconclusive or exhaustive evaluation of the subject matter's safety, functionality, or reliability. Stakeholders are encouraged to conduct their own independent reviews, assessments, and validations to ensure the integrity and security of the evaluated subject.

The authors and auditors of this report disclaim any liability for any direct, indirect, incidental, or consequential damages or losses that may result from reliance on this report or its contents. It is the responsibility of the stakeholders to ensure the ongoing monitoring and evaluation of the subject matter to mitigate potential risks or vulnerabilities.

The nature of technology and digital systems means that they are inherently subject to risks, including but not limited to vulnerabilities, bugs, and security threats that may not be foreseeable at the time of this report. The dynamic and evolving nature of technological standards, security practices, and threat landscapes means that absolute security cannot be guaranteed. Despite thorough analysis and evaluation, unforeseen vulnerabilities may exist, and new threats may emerge subsequent to the issuance of this report.

The authors and auditors make no guarantee regarding the impenetrability or infallibility of the subject matter under evaluation. Stakeholders are advised to implement comprehensive security measures, including but not limited to regular updates, patches, and monitoring, to safeguard against potential threats and vulnerabilities.