BURRA SEC

PancakeSwap Security Review

Reviewed by: Goran Vladika

13th May - 15 May, 2025

PancakeSwap Security Review Report

Burra Security

May 15, 2025

Introduction

A time-boxed security review of the **PancakeSwap** protocol was done by **Burra Security** team, focusing on the security aspects of the smart contracts.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource, and expertise-bound effort where we try to find as many vulnerabilities as possible. We can not guarantee 100% security after the review or even if the review will find any vulnerabilities. Subsequent security reviews, bug bounty programs, and on-chain monitoring are recommended.

About Burra Security

Burra Sec offers security auditing and advisory services with a special focus on cross-chain and interoperability protocols and their integrations.

About PancakeSwap

PancakeSwap's cross-chain contracts allow token exchanges between blockchains by integrating on-chain swaps with cross-chain bridging. They support swap→bridge, bridge→swap, and swap→bridge→swap flows, leveraging the Across Protocol for bridging.

Severity classification

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

Impact - The technical, economic, and reputation damage from a successful attack

Likelihood - The chance that a particular vulnerability gets discovered and exploited

Severity - The overall criticality of the risk

Informational - Findings in this category are recommended changes for improving the structure, usability, and overall effectiveness of the system.

Security Assessment Summary

***review commit hash* - 4b136c1603409c97c80c01534437f2bc614fa89**

Scope

The following smart contracts were in the scope of the audit:

- src/Dispatcher.sol
- src/XChainSender.sol
- src/adapters/AcrossAdapter.sol
- src/base/ReentrancyGuardTransient.sol
- src/libraries/Commands.sol
- src/libraries/Constants.sol
- src/libraries/LibAddress.soLSt
- src/libraries/PCSOrder.sol
- src/libraries/Payments.sol

Findings Summary

ID	Title	Severity	Status
M-01	Positive slippage captured by the relayer instead of the user	Medium	Fixed

Detailed Findings

[M-01] Positive slippage captured by the relayer instead of the user

Target

- `Dispatcher.sol#L28`

Severity

- Impact: Low
  - Users are guaranteed to lose any positive slippage above their `minOutputAmount`, allowing the relayer to pocket undeserved gains.
- Likelihood: High
  - This issue will occur on every swap+bridge operation unless the swap executes exactly at the worst-case price.

Description

The protocol allows users to perform a swap on the source chain followed by a bridging action to the destination chain through AcrossV3. In the calldata, user must provide the minimum acceptable swap execution amount (minOutputAmount), as well as the input and output amounts for the AcrossV3 bridge.

Here's the issue: the user cannot predict the actual swap execution price ahead of time. To avoid transaction reverts, they must assume the worst-case scenario and set the bridge input/output values based on the minimum expected swap execution price. Although the contract dynamically overwrites `bridgeData.inputAmount` with the actual post-swap token balance, it leaves `bridgeData.outputAmount` fixed to the user's pre-calculated worst-case value. As a result, any positive slippage

above `minOutputAmount` is captured by the relayer instead of the user, leading to guaranteed financial loss whenever the swap executes above the minimum threshold.

As an example - user wants to swap 1 ETH for USDC on the source chain, then bridge all received USDC. Let's say for simplicity 1 ETH = 1000 USDC, relayer fee is 1% and user's slippage tolerance is 1%.

User constructs the params to be provided:

- swap data: path eth->usdc, minOutputAmount=990 (1% slippage tolerance)
- bridge data: inputAmount=990 (min swap result), outputAmount=980.1 (1% relayer fee on top of min swap result)

Market conditions change at a time of the swap execution and swap results in 1010 USDC. The contract updates `inputAmount` to 1010, but `outputAmount` remains 980.1. Those values are provided to the AcrossV3. So as a result:

- user receives 980.1 USDC
- relayer pocketed 29.9 USDC (~3% fee)

If `outputAmount` had been re-calculated as 99% of the actual post-swap amount:

- user would receive 999.9 USDC
- relayer would get 10.1 USDC (1% according to fair relayer fee)

So the user effectively lost `999.9 - 980.1 = 19.8 USDC`. This kind of loss is inevitable unless the swap executes exactly at the `minOutputAmount`. The root cause is that `bridgeData.inputAmount` is dynamic (set post-swap), while `bridgeData.outputAmount` is static (pre-set before the swap outcome is known).

The loss for a user in a single transaction would correspond to their slippage tolerance — around 0.5% to 1% of the trade value. Over time total amount of lost funds by all the protocol users could grow to millions of dollars.

Recommendation

Since the user-provided `outputAmount` is based on the worst-case swap execution, the protocol should dynamically scale it based on the actual post-swap result. If the swap executes at a rate X% better than the minimum, then `outputAmount` should be increased proportionally before initiating the bridge.

PancakeSwap

Fixed in PR#88.

BurraSec

Fix looks good.