Defi Plaza

StablePlaza Security Analysis

by Pessimistic

This report is public

July 1, 2022

Abstract

In this report, we consider the security of smart contracts of StablePlaza project. Our task is to find and describe security issues in the smart contracts of the platform.

Disclaimer

The audit does not give any warranties on the security of the code. A single audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, a security audit is not investment advice.

Summary

In this report, we considered the security of StablePlaza smart contracts. We performed our audit according to the procedure described below.

The initial audit identified only a few issues of low severity. After the initial audit the codebase was updated. The developers fixed all of the issues from the initial audit.

Later, the codebase was updated again, with improved NatSpec comments and minor protocol changes.

After that, the codebase was updated again, with a fix for staking reward calculation.

The overall code quality of the project is good.

General recommendations

We recommend adding public documentation to the project.

Project overview

Project description

For the audit, we were provided with StablePlaza project on a private repository, commit d57aeac209afb0a90b28017b3e3c05e9baa24fa4.

The code has detailed NatSpec comments. However, there is no detailed documentation available. The project whitepaper is under development.

98/98 tests pass.

The total LOC of audited sources is 386.

Codebase update

After the initial audit, the codebase was updated. For the recheck, we were provided with commit a06dfd083d077eb4af41218cab57a5d0eebbccc0.

The developers addressed all the issues and refactored the codebase. New functionality for liquidity handling was added. No new issues were found.

Also, the developers added new tests to the project; the overall test count is 163. All tests pass. The source code is fully covered with tests.

Codebase update #2

Later, the codebase was updated again. For the recheck, we were provided with commit a9c2469c4342b169003f3b617194fc4c3bea2286.

This update includes only minor changes to the protocol, simplifying the economic implementation. The liquidity adding fee was removed; as a side-effect of such simplification, a penalty for withdrawing liquidity just after adding appeared (see comments for N02). No new issues were found.

Codebase update #3

After that, the codebase was updated again. For the recheck, we were provided with commit 5bdd58234f23a496084d09de17903eb6531ad677.

This update fixes a low severity bug that occurred in the case of recurring staking (contract performed inaccurate rewards calculation).

Procedure

In our audit, we consider the following crucial features of the code:

1. Whether the code is secure.
2. Whether the code corresponds to the documentation (including whitepaper).
3. Whether the code meets best practices.

We perform our audit according to the following procedure:

- Automated analysis
  - We scan the project's codebase with the automated tools: Slither.
  - We manually verify (reject or confirm) all the issues found by the tools.
- Manual audit
  - We manually analyze the codebase for security vulnerabilities.
  - We assess the overall project structure and quality.
- Report
  - We reflect all the gathered information in the report.

Manual analysis

The contracts were completely manually analyzed, their logic was checked. Besides, the results of the automated analysis were manually verified. All the confirmed issues are described below.

Critical issues

Critical issues seriously endanger project security. They can lead to loss of funds or other catastrophic consequences. The contracts should not be deployed before these issues are fixed.

The audit showed no critical issues.

Medium severity issues

Medium issues can influence project operation in the current implementation. Bugs, loss of potential income, and other non-critical failures fall into this category, as well as potential problems related to incorrect system management. We highly recommend addressing them.

The audit showed no issues of medium severity.

Low severity issues

Low severity issues do not directly affect project operation. However, they might lead to various problems in future versions of the code. We recommend fixing them or explaining why the team has chosen a particular option.

L01. Locked tokens (fixed)

Consider adding a function to the **StablePlaza** contract to withdraw accidental tokens from the contracts balance.

The issue has been fixed and is not present in the latest version of the code.

L02. Code quality (fixed)

There is code duplication in the **StablePlaza** contract at lines 136-146 and 158-168. Consider introducing an internal function with common functionality to minimize possible errors.

The issue has been fixed and is not present in the latest version of the code.

L03. Events (fixed)

The `changeListedToken`, `setAdmin` and `updateConfig` functions of the **StablePlaza** contract change the key parameters of the protocol. Consider emitting corresponding events for these cases.

The issue has been fixed and is not present in the latest version of the code.

L04. Dependency management (fixed)

The project configuration file (**brownie-config.yaml**) does not include `dependencies` section. Consider explicitly specifying `openzeppelin-contracts@4.5.0` in this section.

The issue has been fixed and is not present in the latest version of the code.

Notes

N01. Code quality (fixed)

Consider utilizing underscore notation (i.e., `10_000_000` instead of `10000000`) to improve readability and minimize possible errors in the **StablePlaza** contract.

The issue has been fixed and is not present in the latest version of the code.

N02. Liquidity mining (addressed)

Since liquidity providers receive their income as part of the protocol fee, it takes time (protocol transactions) to make a profit. Note that adding liquidity to the protocol and removing it immediately thereafter might result in a minor loss (up to ~0.6%).

> *Comment from the developers: It was noted that adding liquidity and immediately removing it again could lead to a slight loss to the user. This is indeed the case, and is caused by liquidity adding fee (0.03%) and approximation error for large amounts of liquidity in the simplified math. A note was made in the code to explicitly reflect this. The approximation error is near zero for small liquidity amounts and is limited to ~0.6% for the maximum amounts of liquidity add possible (more than 1.5 times the total liquidity currently present in the exchange).*

> *In the lastest version of the code the liquidity adding fee was removed.*

> *Comment from the developers: Adding followed by withdrawing incurs a penalty of ~0.74% when the exchange is in balance. The penalty can be mitigated or even be converted into a gain by adding to a token that is underrepresented and withdrawing from a token that is overrepresented in the exchange.*

N03. Possible loss for liquidity providers

The project achieves high capital efficiency via delta logic. However, the liquidity providers are exposed to a much larger impermanent loss as a trade-off. It might lead to high losses in case of prolonged or permanent depeg of one of the stabletokens.

> *Comment from the developers: If any of the coins falls below peg for more than 2% and stays off-peg, the exchange will end up holding only this token due to arbitrage. This will indeed cause a loss to the liquidity providers. In such a case they can either take the loss and withdraw their liquidity, or they can wait until the token regains peg within 2% of the other listed tokens.*

N04. Locking response time sensitivity

Since price volatility might result in the arbitrage possibility for profitable swaps, the admin may not have time to lock the pool and this will lead to a large increase in the amount of cheap tokens. We recommend implementing quick emergency lock mechanism for such cases. E.g, the possibility to lock the exchange not only by admin.

> *Comment from the developers: Due to the close price range for trading stable tokens it may indeed be difficult to timely lock the exchange in time when an issue arises with one of the listed tokens. The team is looking at implementing an automated exchange lock in case certain metrics are violated. This can be done from the existing code base so it shouldn't be an issue technically, we just need to agree with the community when the exchange should be locked.*

This analysis was performed by Pessimistic:

Evgeny Marchenko, Senior Security Engineer
Vladimir Pomogalov, Security Engineer
Vladimir Tarasov, Security Engineer
Irina Vikhareva, Project Manager

July 1, 2022