XYZVERSE × PESSIMISTIC

SECURITY ANALYSIS

by Pessimistic

This report is public

October 3, 2024

ABSTRACT

In this report, we consider the security of smart contracts of XYZVERSE Presale 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 XYZVERSE Presale smart contracts. We described the audit process in the section below.

The audit showed one issue of medium severity: Project roles. Also, one low-severity issue was found.

The overall quality of the code is good.

GENERAL RECOMMENDATIONS

We do not have any recommendations for the project.

PROJECT OVERVIEW

Project description

For the audit, we were provided with XYZVERSE Presale project on a private GitHub repository, commit e9acca38bc34e0579362893bed57a9dc5e1a9a52.

The scope of the audit included **XYZPresale.sol** and **IXYZPresale.sol** files.

The documentation for the project included NatSpec comments, as well as short description from the developers.

All 31 tests pass successfully. The code coverage is 96.49%.

The total LOC of audited sources is 138.

AUDIT PROCESS

We started the audit on September 18, 2024 and finished on September 19, 2024.

We inspected the materials provided for the audit and decided to pay extra attention to the following parts of the code:

- The correctness of the conversion formula;
- Chainlink integration implementation;
- The presence of the bugs in the code.

We manually analyzed all the contracts within the scope of the audit and checked their logic. In addition, we scanned the project with the following tools:

- Static analyzer Slither;
- Our plugin Slitherin with an extended set of rules,
- Semgrep rules for smart contracts.

We ran tests and calculated the code coverage. After that, we combined in a private report all the verified issues we found during the manual audit or discovered by automated tools.

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 severity 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.

M01. **Project roles**

The project design is heavily relied on the `Owner` role. It has the following powers:

- Pause and unpause token purchases;
- Change the `treasury` address;
- Add any amount of tokens to an arbitrary address;
- Add or remove payment token;
- Update purchasing price. Note, that the result of an unexpected malicious price change can be mitigated by introducing the minimum return amount argument to the purchasing function.

Since the system depends heavily on the owner, there are scenarios that can lead to undesirable consequences for the project and its users, e.g., if the owner's private key becomes compromised.

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. Check the timestamp of the latest price answer

In the **XYZPresale** contract, the `purchaseWithNativeCoin` function obtains the price of the native coin from the chainlink price feed. While it is expected that the price feed for the native currency is regularly updated, we still recommend checking the timestamp of the latest answer in case of unexpected downtimes.

Notes

N01. Not all stablecoins are supported

Note that the **XYZPresale** contract works only with the stablecoins that have 6 decimals value. This might not be the case for stablecoins on some chains (for example, USDC token on the BNB chain).

N02. Price difference for stablecoins

The **XYZPresale** contract assumes that the price of stablecoins is 1$. We recommend monitoring the actual prices of the used tokens so that they differ slightly and are not depeged.

This analysis was performed by Pessimistic:

Pavel Kondratenkov, Senior Security Engineer
Yhtyyar Sahatov, Security Engineer
Konstantin Zherebtsov, Business Development Lead
Irina Vikhareva, Project Manager
Alexander Seleznev, CEO

October 3, 2024