SwarmMarkets SX1411 Child Token Security Analysis

by Pessimistic

This report is private

March 4, 2022

Abstract

In this report, we consider the security of smart contracts of SwarmMarkets SX1411 Child Token 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 SwarmMarkets SX1411 Child Token smart contracts. We performed our audit according to the procedure described below.

The audit showed two issues of medium severity, including an Overpowered role and an Issue with tests. Also, one low-severity issue was found.

General recommendations

We recommend fixing the mentioned issues. We also recommend implementing CI to run tests, calculate code coverage, and analyze code with linters and security tools.

Project overview

Project description

For the audit, we were provided with SwarmMarkets SX1411 Child Token project on a private GitHub repository, commit 0873a05f0aa590272c595327020e75b9a80707f5.

The scope of the audit included only the following files:

- **AccessManagerChild.sol**
- **AssetTokenChild.sol**

The documentation for the project was provided as a google document. The code has detailed NatSpec comments.

347 tests out of 348 pass successfully, one test fails. The code coverage for the scope of the audit is 99.2%.

The total LOC of audited sources is 265.

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 and SmartCheck.
  - 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.

M01. Overpowered roles

The `Admin` role can:

- Grant roles: `DEFAULT_ADMIN_ROLE`, `SAFEGUARD_TRANSFER_ROLE`, `EXECUTOR_ROLE`.
- Change `childChainManagerProxy` address.
- Add/remove agents.
- Add/remove members to/from blacklist.

The `Admin` role can:

- Freeze/unfreeze contract.

The `Admin` role can:

- Add/remove addresses to/from Authorization list.

In the current implementation, the system depends heavily on these roles. Thus, there are scenarios that can lead to undesirable consequences for the project and its users, e.g., if private keys for any of these roles become compromised.

We recommend designing contracts in a trustless manner or implementing proper key management, e.g., setting up a multisig.

M02. Tests issues

One test out of 348 fails to pass.

Testing is crucial for code security, and an audit does not replace tests in any way. Even a single failing test indicates that probably the code is not working properly. All tests must pass. All important scenarios should be covered with tests.

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. Code quality

In **AccessManagerChild** contract, `onlyAdmin` modifier is redundant in `grantAdminRole`, `grantSafeguardTransferRole`, and `grantExecutorRole` functions. These functions call `grantRole` function that already has `onlyAdmin` modifier.

In this case, we recommend calling the `grantRole` function directly.

This analysis was performed by Pessimistic:

Daria Korepanova, Security Engineer
Evgeny Marchenko, Senior Security Engineer
Nikita Kirillov, Junior Security Engineer
Boris Nikashin, Analyst
Irina Vikhareva, Project Manager

March 4, 2022