Synthetix Alpheratz Release Smart Contract Audit

1. Introduction

iosiro was commissioned by Synthetix to conduct a smart contract audit of their Alpheratz Release, which included the following components:

- SIP-148 - audited over several iterations starting from 17 to 25 March 2022 with two auditors, continuing on 22 April 2022 with one auditor, and concluding from 12 to 16 May 2022 with two auditors, consuming a total of 18 resource days.
- SIP-236 - audited on 13 May 2022 with two auditors, consuming a total of 2 resource days.

This report is organized into the following sections.

- **Section 2 - Executive summary:** A high-level description of the findings of the audit.

- **Section 3 - Audit details:** A description of the scope and methodology of the audit.
- **Section 4 - Design specification:** An outline of the intended functionality of the smart contracts.
- **Section 5 - Detailed findings:** Detailed descriptions of the findings of the audit.

The information in this report should be used to understand the smart contracts' risk exposure better and as a guide to improving the security posture of the smart contracts by remediating issues identified. The results of this audit reflect the in-scope source code reviewed at the time of the audit.

The purpose of this audit was to achieve the following:

- Identify potential security flaws.
- Ensure that the smart contracts function according to the documentation provided.

Assessing the off-chain functionality associated with the contracts, for example, backend web application code, was outside of the scope of this audit.

Due to the unregulated nature and ease of transfer of cryptocurrencies, operations that store or interact with these assets are considered high risk from cyber attacks. As such, the highest level of security should be observed when interacting with these assets. This requires a forward-thinking approach, which takes into account the new and experimental nature of blockchain technologies. Strategies that should be used to encourage secure code development include:

- Security should be integrated into the development lifecycle, and the level of perceived security should not be limited to a single code audit.
- Defensive programming should be employed to account for unforeseen circumstances.
- Current best practices should be followed where possible.

2. Executive summary

This report presents the findings of a smart contract audit performed by iosiro of Synthetix's Alpheratz release.

SIP-148 introduced a new liquidation mechanism that made the system more resistant to cascading liquidations. The new design redistributes the liquidated debt and collateral to system stakers without requiring the liquidator to burn synths. This allowed collateral to remain in the system as escrowed rewards and mitigated the risk of SNX flooding the market during a cascading liquidation, further dropping the price. Liquidatable accounts can either be self-liquidated by the account owner or can be flagged by a keeper bot. If a flagged account can still be liquidated after some period of time, the account becomes open for liquidation. Flaggers and liquidators earn a fixed SCCP-configurable fee for performing these actions.

One medium-risk issue and two informational issues were identified during the audit; the high-risk issue was closed during the assessment. The code of this system underwent significant changes over the course of the audit, which both simplified the code and improved its quality. The code was of a high standard by the end of the audit.

SIP-236 corrected exchange logic to use the post-settlement amounts when performing exchange calculations.

No issues were identified in SIP-236.

3. Audit details

3.1 Scope

The source code considered in-scope for the assessment is described below. Code from all other files was considered to be out-of-scope. Out-of-scope code that interacts with in-scope code was assumed to function as intended and not introduce any functional or security vulnerabilities for the purposes of this audit.

3.1.1 Synthetix SIP-148 smart contracts

Project Name: Synthetix
Initial commit: 61321b8

Final commit: 02f6845

Files: contracts/Liquidator.sol, contracts/LiquidatorRewards.sol, contracts/Issuer.sol, contracts/BaseSynthetix.sol, contracts/MixinSystemSettings.sol, contracts/SystemSettings.sol, contracts/SystemSettingsLib.sol

3.1.2 Synthetix SIP-236 smart contracts

Project Name: Synthetix

Initial commit: 2f46009

Final commit: 2f46009

Files: contracts/Exchanger.sol, contracts/ExchangerWithFeeRecAlternatives.sol

3.2 Methodology

A variety of techniques were used in order to perform the audit. These techniques are briefly described below.

3.2.1 Code review

The source code was manually inspected to identify potential security flaws. Code review is a useful approach for detecting security flaws, discrepancies between the specification and implementation, design improvements, and high-risk areas of the system.

3.2.2 Dynamic analysis

The contracts were compiled, deployed, and tested in a test environment, both manually and through the test suite provided. Manual analysis was used to confirm that the code was functional and to identify security issues that could be exploited.

3.2.3 Automated analysis

Tools were used to automatically detect the presence of several types of security vulnerabilities, including reentrancy, timestamp dependency bugs, and transaction-ordering dependency bugs. Static analysis results were reviewed manually and any false positives were removed. Any true positive results are included in this report.

Static analysis tools commonly used include Slither, Securify, and MythX. Tools such as the Remix IDE, compilation output, and linters could also be used to identify potential areas of concern.

3.3 Risk ratings

Each issue identified during the audit has been assigned a risk rating. The rating is determined based on the criteria outlined below.

- **High risk** - The issue could result in a loss of funds for the contract owner or system users.
- **Medium risk** - The issue resulted in the code specification being implemented incorrectly.
- **Low risk** - A best practice or design issue that could affect the security of the contract.
- **Informational** - A lapse in best practice or a suboptimal design pattern that has a minimal risk of affecting the security of the contract.
- **Closed** - The issue was identified during the audit and has since been satisfactorily addressed, removing the risk it posed.

4. Design specification

The following section outlines the intended functionality of the system at a high level.

4.1 SIP-148

The specification of SIP-148 was based on commit hash 180f4d3.

4.2 SIP-236

The specification of SIP-236 was based on commit hash 46938e7.

5. Detailed findings

The following section includes in-depth descriptions of the findings of the audit.

5.1 High risk

No high-risk issues identified during the audit were present at the conclusion of the audit.

5.2 Medium risk

No medium-risk issues identified during the audit were present at the conclusion of the audit.

5.3 Low risk

No low-risk issues identified during the audit were present at the conclusion of the audit.

5.4 Informational

5.4.1 Distribution of liquidator fees

SIP-148

Description

When the total redeemed amount from a liquidation is less than the sum of the flagger and liquidator fees, the system sends all the redeemed SNX to the stakers and does not pay the flagger or liquidator. This approach does not adequately incentivize liquidators to perform liquidations against small positions, which would leave them open. These small positions could accumulate as bad debt and jeopardize system health. It should be considered that the system should rather prioritize paying the liquidator and flagger fees, instead of the stakers. In this case, liquidator fees can be given preference as it is the more expensive action.

5.4.1 Liquidation flag not removed from healthy positions

SIP-148

Description

When attempting to liquidate a flagged account that is currently healthy, the flag should be removed instead of reverting to prevent other liquidators from trying to liquidate the account. This will also reset the liquidation timer for the account to avoid the account becoming immediately liquidatable in the future.

5.5 Closed

5.5.1 Incorrect debt cache update (medium risk)

SIP-148 Issuer.sol#L672

Description

During liquidation, the system incorrectly subtracted the liquidated debt from the debt cache, misrepresenting the system value on that layer. This discrepancy would be corrected during the next debt snapshot.

Update

Removed in 152b6df.