Synthetix Alnitak Release Smart Contract Audit

1. Introduction

iosiro was commissioned by Synthetix to conduct a smart contract audit of the implementation of SIP-138, SIP-139, SIP-140, and SIP-151 for the Alnitak Release. The audit was performed by one auditor on 21 May 2021, consuming a total of 1 resource days. A review was performed on 25 May 2021 and on 24 June 2021.

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 risk exposure of the smart contracts, and as a guide to improving the security posture of the smart contracts by remediating the issues that were identified. The results of this audit are only a reflection of the source code reviewed at the time of the audit and of the source code that was determined to be in-scope.

The purpose of this audit was to achieve the following:

- Ensure that the smart contracts functioned as intended.
- Identify potential security flaws.

Assessing the market effect, economics, game theory, or underlying business model of the platform were strictly beyond 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 very high risk with regards to 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. There are a number of techniques that can help to achieve this, some of which are described below.

- Security should be integrated into the development lifecycle.
- Defensive programming should be employed to account for unforeseen circumstances.
- Current best practices should be followed when possible.

2. Executive Summary

This report presents the findings of the audit performed by iosiro of the smart contract implementation of the Alnitak release.

The purpose of SIP-138 was to improve volume tracking on L1 and L2 by adding a fee parameter to the `ExchangeTracking` event. No issues were identified during the audit.

Overall, the implementation was of a high standard and accorded with the specification provided.

The purpose of SIP-139 was to add a protected function to reset the last price of an exchange for synths after a circuit breaker reset from SIP-65. One low risk issue was identified and fixed during the audit. Overall, the implementation was of a high standard and accorded with the specification provided.

The purpose of SIP-140 was to add an additional exchange function to support transactions initiated by a user, but performed through another address. One low risk issue was identified during the audit. Overall, the implementation was of a high standard and accorded with the specification provided.

The purpose of SIP-151 was to programmatically generate solidity contracts to perform atomic upgrades of the Synthetix system. Upgrades previously required the system to be paused to prevent any data corruption. Upgrades also often consist of performing a number of owner actions that needed to be sequentially executed. By deploying a contract with all the owner actions in a single function, the upgrade process can be simplified while maintaining the overall security architecture. No issues were identified during the audit. Overall, the implementation was of a high standard and accorded with the specification provided.

3. Audit Details

3.1 Scope

The source code considered in-scope for the assessment is described below. Code from all other files is considered to be out-of-scope. Out-of-scope code that interacts with in-scope code is assumed to function as intended and introduce no functional or security vulnerabilities for the purposes of this audit.

3.1.1 Synthetix SIP-138 Smart Contracts

Project Name: Synthetix
Commits: aeb11d9
Files: contracts/BaseSynthetix.sol, contracts/Exchanger.sol

3.1.2 Synthetix SIP-139 Smart Contracts

Project Name: Synthetix
Commits: 8b766d2, 500c84f
Files: contracts/Exchanger.sol

3.1.3 Synthetix SIP-140 Smart Contracts

Project Name: Synthetix
Commits: 06ce310, 00df930, f141ab9
Files: contracts/BaseSynthetix.sol, contracts/PurgeableSynth.sol, contracts/Synth.sol, contracts/Synthetix.sol

3.1.4 Synthetix SIP-151 Smart Contracts

Project Name: Synthetix
Commits: 75e51e7
Files: contracts/BaseMigration.sol, contracts/legacy/LegacyOwned.sol, contracts/legacy/LegacyTokenState.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 operated at a functional level, and to verify the exploitability of any potential security issues identified.

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. The static analysis results were manually analyzed to remove false-positive results. True positive results would be indicated in this report. Static analysis tools commonly used include Slither, Securify, and MythX. Tools such as the Remix IDE, compilation output, and linters are also used to identify potential issues.

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 addressed to a satisfactory level to remove the risk that it posed.

4. Design Specification

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

4.1 SIP-138

The specification of SIP-138 was based on commit hash b7813321.

4.2 SIP-139

The specification of SIP-139 was based on commit hash b7813321.

4.3 SIP-140

The specification of SIP-140 was based on commit hash b7813321.

4.4 SIP-151

The specification of SIP-151 was based on commit hash 9847b1c.

5. Detailed Findings

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

5.1 High Risk

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

5.2 Medium Risk

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

5.3 Low Risk

5.3.1. Potentially dangerous integration with third-party applications

SIP-140

Description

The implementation of `exchangeWithTrackingForInitiator` required that the function use `tx.origin` as the destination address of the exchange performed. As a result, certain third-party applications, such as multi-signature wallets, may be incompatible with the function. For example, if using Gnosis Safe, the user who performs the transaction after all parties have signed will receive the funds instead of the wallet itself.

The risk is mitigated by the fact that it is a specialized exchange function, therefore contracts would need to deliberately interact with it. The risk of this issue was identified and accepted by the Synthetix team.

5.4 Informational

No informational issues were identified.

5.5 Closed

5.5.1 Exchange rates are not validated (low risk)

SIP-139: Exchanger.sol#L663

Description

The `resetLastExchangeRate` function did not verify whether the exchange rates were valid at the time of being reset. As a result, it would be possible for the admin to reset a rate to an invalid state.

Recommendation

It is recommended that the `exchangeRates.anyRateIsInvalid(...)` function is used to determine whether the rates are valid before using them to update the last exchange rate.

The risk of this issue was mitigated by the fact that the owner could independently verify rates on-chain before resetting them.

Update

Implemented in 500c84f.

5.5.2 Unnecessary validation (informational)

SIP-140: Exchanger.sol#L343, Exchanger.sol#L362

Description

The `rewardAddress` was validated to ensure that it was non-zero twice, which unnecessarily consumed gas. In order to improve performance, one of the checks should be removed.

Update

Implemented in f141ab9.

ABOUT
CONTACT US
SMART CONTRACT AUDITING
PENETRATION TESTING
AUDIT REPORTS
PRIVACY POLICY
TERMS OF SERVICE