GUARDIAN

Nunchi

Nunchi SY & Genesis Vaults

Security Assessment

November 22nd, 2025

Summary

Audit Firm Guardian

Prepared By Owen Thurm

Client Firm Nunchi

Final Report Date November 22, 2025

Audit Summary

Nunchi engaged Guardian to review the security of their Nunchi SY & Genesis Vaults. From the 3rd of November to the 5th of November, an auditor reviewed the source code in scope. All findings have been recorded in the following report.

Confidence Ranking

Given the lack of critical issues detected and minimal code changes following the main review, Guardian assigns a Confidence Ranking of 5 to the protocol. Guardian advises the protocol to consider periodic review with future changes. For detailed understanding of the Guardian Confidence Ranking, please see the rubric on the following page.

Guardian Confidence Ranking

Confidence Ranking	Definition and Recommendation	Risk Profile
5: Very High Confidence	Codebase is mature, clean, and secure. No High or Critical vulnerabilities were found. Follows modern best practices with high test coverage and thoughtful design. **Recommendation:** Code is highly secure at time of audit. Low risk of latent critical issues.	0 High/Critical findings and few Low/Medium severity findings.
4: High Confidence	Code is clean, well-structured, and adheres to best practices. Only Low or Medium-severity issues were discovered. Design patterns are sound, and test coverage is reasonable. Small changes, such as modifying rounding logic, may introduce new vulnerabilities and should be carefully reviewed. **Recommendation:** Suitable for deployment after remediations; consider periodic review with changes.	0 High/Critical findings. Varied Low/Medium severity findings.
3: Moderate Confidence	Medium-severity and occasional High-severity issues found. Code is functional, but there are concerning areas (e.g., weak modularity, risky patterns). No critical design flaws, though some patterns could lead to issues in edge cases. **Recommendation:** Address issues thoroughly and consider a targeted follow-up audit depending on code changes.	1 High finding and ≥ 3 Medium. Varied Low severity findings.
2: Low Confidence	Code shows frequent emergence of Critical/High vulnerabilities (~2/week). Audit revealed recurring anti-patterns, weak test coverage, or unclear logic. These characteristics suggest a high likelihood of latent issues. **Recommendation:** Post-audit development and a second audit cycle are strongly advised.	2-4 High/Critical findings per engagement week.
1: Very Low Confidence	Code has systemic issues. Multiple High/Critical findings (≥5/week), poor security posture, and design flaws that introduce compounding risks. Safety cannot be assured. **Recommendation:** Halt deployment and seek a comprehensive re-audit after substantial refactoring.	≥5 High/Critical findings and overall systemic flaws.

Table of Contents

Project Information

Smart Contract Risk Assessment

Addendum

Project Overview

Project Summary

Project Name	Nunchi
Language	Solidity
Codebase	https://github.com/Nunchi-trade
Commit(s)	Main Review commit: e03566d054e26442e9ff324aff4cd34e186227bd Remediation Review commit: 697f05b70cdb5d73c9b6f9b9427d6a7cccfdd162

Audit Summary

Delivery Date	November 22, 2025
Audit Methodology	Static Analysis, Manual Review, Test Suite, Contract Fuzzing

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● High	2	0	0	0	0	2
● Medium	0	0	0	0	0	0
● Low	4	0	0	1	0	3
● Info	0	0	0	0	0	0

Audit Scope & Methodology

Scope and details:

This audit will cover the code changes introduced in the following diff:

https://github.com/Nunchi-trade/genesis-vaults/compare/417a877e2472e126af998ec1518871f512b6e1b6..e03566d054e26442e9ff324aff4cd34e186227bd#diff-4992d8303f33c4ad95d7c58ca462a4120f0b1d596b3c260842905d4a90059efc

Specifically:

Modifications within the src/genesis directory

The newly introduced contract at src/hyperbeat-vault/NunchiHyperbeatVaultSY.sol

Audit Scope & Methodology

Vulnerability Classifications

Severity	Impact: *High*	Impact: *Medium*	Impact: *Low*
Likelihood: *High*	🔴 Critical	🟠 High	🟡 Medium
Likelihood: *Medium*	🟠 High	🟡 Medium	🟢 Low
Likelihood: *Low*	🟡 Medium	🟢 Low	🟢 Low

Impact

High Significant loss of assets in the protocol, significant harm to a group of users, or a core functionality of the protocol is disrupted.

Medium A small amount of funds can be lost or ancillary functionality of the protocol is affected. The user or protocol may experience reduced or delayed receipt of intended funds.

Low Can lead to any unexpected behavior with some of the protocol's functionalities that is notable but does not meet the criteria for a higher severity.

Likelihood

High The attack is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount gained or the disruption to the protocol.

Medium An attack vector that is only possible in uncommon cases or requires a large amount of capital to exercise relative to the amount gained or the disruption to the protocol.

Low Unlikely to ever occur in production.

Audit Scope & Methodology

Methodology

Guardian is the ultimate standard for Smart Contract security. An engagement with Guardian entails the following:

- Two competing teams of Guardian security researchers performing an independent review.
- A dedicated fuzzing engineer to construct a comprehensive stateful fuzzing suite for the project.
- An engagement lead security researcher coordinating the 2 teams, performing their own analysis, relaying findings to the client, and orchestrating the testing/verification efforts.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts. Comprehensive written tests as a part of a code coverage testing suite.
- Contract fuzzing for increased attack resilience.

Findings & Resolutions

ID	Title	Category	Severity	Status
H-01	exchangeRate Doesn’t Follow ERC5115 Standard	Unexpected Behavior	● High	Resolved
H-02	Incorrect Interface Bricks The System	DoS	● High	Resolved
L-01	Native Value Not Handled	Validation	● Low	Resolved
L-02	getTokensIn Is Hardcoded	Configuration	● Low	Acknowledged
L-03	No Rescue Functionality	Warning	● Low	Resolved
L-04	PreviewDeposit Misleading Result	Warning	● Low	Resolved

H-01 | exchangeRate Doesn’t Follow ERC5115 Standard

Category	Severity	Location	Status
Unexpected Behavior	● High	NunchiHyperbeatVaultSY.sol	Resolved

Description

As described in the ERC 5115 ERC document: https://eips.ethereum.org/EIPS/eip-5115
The exchangeRate function MUST return ExchangeRate(t_now) such that ExchangeRate(t_now) * syBalance / 1e18 = assetBalance.

And, that the exchangeRate method updates and returns the latest exchange rate, which is the exchange rate from SY token amount into asset amount, scaled by a fixed scaling factor of 1e18.

The PRICER contract however returns a rate with 8 decimals of precision, and the SY token balances are using 6 decimals as derived from the vault token decimals of 6.

Therefore the exchangeRate function is non-compliant and will almost always round to zero if adjusted by a scaling factor of 1e18.

Recommendation

Consider scaling the exchangeRate result by 1e10 to be compliant with the ERC 5115 standard.

Resolution

Nunchi Team: The issue was resolved in PR#26.

H-02 | Incorrect Interface Bricks The System

Category	Severity	Location	Status
DoS	● High	Global	Resolved

Description

The IDepositor interface includes a deposit function interface that is supposed to include returndata for a uint256 value, however the implementation of the DEPOSITOR at the address 0x0868A605661440e5D58453f16BDB64795B2Da176 on HyperEVM does not return anything.

This ultimately results in a decoding revert as no return data was provided while some was expected, thus DoSing the entire deposit flow.

Recommendation

Remove the uint256 return value from the IDepositor interface.

Resolution

Nunchi Team: The issue was resolved in PR#26.

L-01 | Native Value Not Handled

Category	Severity	Location	Status
Validation	● Low	SYBaseUpgV2.sol	Resolved

Description

The deposit function in the SYBaseUpgV2 contract is payable, however the overridden _deposit function in the NunchiHyperbeatVaultSY contract does not handle msg.value.

Recommendation

Add a validation in the _deposit function to revert if nonzero msg.value has been provided.

Resolution

Nunchi Team: The issue was resolved in PR#26.

L-02 | getTokensIn Is Hardcoded

Category	Severity	Location	Status
Configuration	● Low	NunchiHyperbeatVaultSY.sol	Acknowledged

Description

getTokensIn() returns a static array [VAULT_TOKEN, USDC, USDT0] for simplicity while deposit validation uses DEPOSITOR.isDepositToken().

If DEPOSITOR changes supported tokens, integrators relying on getTokensIn may route deposits to unsupported tokens and revert, degrading UX and composability.

Recommendation

Keep getTokensIn synchronized with DEPOSITOR.isDepositToken, or remove/hard-deprecate the static list and instead expose a function that queries the depositor's current set. Document that getTokensIn is non-authoritative if kept.

Resolution

Nunchi Team: Acknowledged.

L-03 | No Rescue Functionality

Category	Severity	Location	Status
Warning	● Low	NunchiHyperbeatVaultSY.sol	Resolved

Description

The SYBaseUpgV2 contract implements a receive function, however there is no rescue logic to recover any Ether that may have been accidentally sent to the contract.

Recommendation

Consider if a rescue function should be implemented.

Resolution

Nunchi Team: The issue was resolved in PR#26.

L-04 | PreviewDeposit Misleading Result

Category	Severity	Location	Status
Warning	● Low	Global	Resolved

Description

The previewDeposit function does not take into account the deposit cap that is implemented on the underlying Depositor contract, and as a result this can be misleading for any consumers of this function.

Recommendation

Consider checking the Depositor for the deposit cap to surface to the user whether the requested deposit will fail.

Resolution

Nunchi Team: The issue was resolved in PR#26.

Disclaimer

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Guardian to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Guardian’s position is that each company and individual are responsible for their own due diligence and continuous security. Guardian’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by Guardian is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract’s safety and security. Therefore, Guardian does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

About Guardian

Founded in 2022 by DeFi experts, Guardian is a leading audit firm in the DeFi smart contract space. With every audit report, Guardian upholds best-in-class security while achieving our mission to relentlessly secure DeFi.

To learn more, visit https://guardianaudits.com

To view our audit portfolio, visit https://github.com/guardianaudits

To book an audit, message https://t.me/guardianaudits