hexens x STAKEWISE

Security Review Report for StakeWise

August 2025

Table of Contents

1. About Hexens

Hexens is a pioneering cybersecurity firm dedicated to establishing robust security standards for Web3 infrastructure, driving secure mass adoption through innovative protection technology and frameworks. As an industry elite experts in blockchain security, we deliver comprehensive audit solutions across specialized domains, including infrastructure security, Zero Knowledge Proof, novel cryptography, DeFi protocols, and NFTs.

Our methodology combines industry-standard security practices combined with unique methodology of two teams per audit, continuously advancing the field of Web3 security. This innovative approach has earned us recognition from industry leaders.

Since our founding in 2021, we have built an exceptional portfolio of enterprise clients, including major blockchain ecosystems and Web3 platforms.

2. Executive Summary

This audit focused on the update to the StakeWise V3 Periphery contracts. The update introduces a modification to the borrow state calculation to align with the new rounding technique introduced in the latest Aave protocol update.

Our security assessment consisted of a one-day review of the pull request implementing this update.

During the audit, we did not identify any major vulnerabilities. However, we did find one low-severity issue.

The reported issue was fixed by the development team and subsequently validated by us.

We can confidently state that the overall security and code quality have improved following the completion of our audit.

3. Security Review Details

▪ Review Led by

Trung Dinh, Lead Security Researcher

▪ Scope

The analyzed resources are located on:

🔗 https://github.com/stakewise/v3-periphery/pull/17

The issues described in this report were fixed in the following commit:

🔗 https://github.com/stakewise/v3-periphery/pull/17

📌 **Commit:** `7aef6b1e577796a59efa6c341891a2dcc7e010f1`

▪ Changelog

18 August 2025	Audit start
20 August 2025	Initial report
27 August 2025	Revision received
28 August 2025	Final report

4. Severity Structure

The vulnerability severity is calculated based on two components:

1. Impact of the vulnerability
2. Probability of the vulnerability

Impact	Probability			
	Rare	Unlikely	Likely	Very likely
Low	Low	Low	Medium	Medium
Medium	Low	Medium	Medium	High
High	Medium	Medium	High	Critical
Critical	Medium	High	Critical	Critical

- **Severity Characteristics**

Smart contract vulnerabilities can range in severity and impact, and it's important to understand their level of severity in order to prioritize their resolution. Here are the different types of severity levels of smart contract vulnerabilities:

Critical

Vulnerabilities that are highly likely to be exploited and can lead to catastrophic outcomes, such as total loss of protocol funds, unauthorized governance control, or permanent disruption of contract functionality.

High

Vulnerabilities that are likely to be exploited and can cause significant financial losses or severe operational disruptions, such as partial fund theft or temporary asset freezing.

Medium	Vulnerabilities that may be exploited under specific conditions and result in moderate harm, such as operational disruptions or limited financial impact without direct profit to the attacker.
Low	Vulnerabilities with low exploitation likelihood or minimal impact, affecting usability or efficiency but posing no significant security risk.
Informational	Issues that do not pose an immediate security risk but are relevant to best practices, code quality, or potential optimizations.

- **Issue Symbolic Codes**

Each identified and validated issue is assigned a unique symbolic code during the security research stage.

Due to the structure of the vulnerability reporting flow, some rejected issues may be missing.

5. Findings Summary

Severity	Number of findings
Critical	0
High	0
Medium	0
Low	1
Informational	0
Total:	**1**

6. Weaknesses

This section contains the list of discovered weaknesses.

STKW2-1 | Lack of event emission when updating exiting state of the proxy

Fixed

Severity:	Probability:	Impact:
Low	Unlikely	Low

Path:

src/leverage/LeverageStrategy.sol#L486
src/leverage/LeverageStrategy.sol#L420
src/leverage/LeverageStrategy.sol#L286
src/leverage/LeverageStrategy.sol#L308

Description:

The event **StrategyProxyExitingUpdated** is meant to be emitted whenever the state of a strategy proxy changes. In other words, the event should be emitted whenever the value of **isStrategyProxyExiting[proxy]** is updated.

However, the event is only emitted within the function **LeverageStrategy.setStrategyProxyExiting()**, and not in **_enterExitQueue()**, **claimExitedAssets()**, or **rescueVaultAssets()**. This lack of event emission could result in incorrect tracking of the proxy's state on the backend side.

```
/**
 * @notice Event emitted when the strategy proxy is set to exiting state
 * @param proxy The address of the strategy proxy
 * @param isExiting True if the proxy is exiting, false otherwise
 */
event StrategyProxyExitingUpdated(address indexed proxy, bool isExiting);
```

Remediation:

Consider emitting the event when the **isStrategyProxyExiting[proxy]** changes the value.


hexens
x
STAKEWISE