Superfund Smart Contract Audit Report

Naman Blockchain Security

Conducted by: naman (@namx05)

May 9th, 2025 - May 12th, 2025

Contents

1. About

Naman Jain (@namx05) is a smart contract security researcher specializing in EVM, Solana, and CosmWasm ecosystem. He conducts private audits and participates in audit contests. With over 100+ audits across different ecosystems, in which 20+ was Rust audits 🦀 . He has found 150+ Critical/High severity issues.

Naman is dedicated to improving blockchain security. He actively researches and explores ways to make protocols safer. He also writes [articles](https://medium.com/@namx05) to help developers better understand security.

His past work can be found [here](https://github.com/namx05/audits). He can be reached on Twitter [@namx05](https://twitter.com/namx05) or Telegram [@namx05](https://t.me/namx05).

2. Disclaimer

A smart contract security review can't find every vulnerability. It is limited by time, resources, and expertise. The goal is to catch as many issues as possible, but I can't promise complete security. I also can't guarantee that the review will find any problems. To stay safer, it's best to do more reviews, run bug bounty programs, and keep monitoring on-chain activity.

3. Introduction

A time-boxed security review was done by **Naman Jain**, with a focus on the security aspects of the smart contracts implementation. The Superfund team has been very responsive to Naman's inquiries, demonstrating a strong commitment to security by taking into account all the recommendations and fixing suggestions from the researchers.

4. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

3.1 Impact

- High - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- Medium - leads to a moderate material loss of assets in the protocol or moderately harms a group of users.
- Low - leads to a minor material loss of assets in the protocol or harms a small group of users.

3.2 Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.
- Medium - only a conditionally incentivized attack vector, but still relatively likely.
- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

5. Executive Summary

5.1 Protocol Info

Name:	Superfund
Security Review Timeline:	2 Days
Repository:	https://github.com/Superlend/superfund-strategies-public
Review Commit Hash:	f3c31ba291784774206cce54a6ad2f4a7a932740
Fix Review Commit Hash:	61c7d7c7a8fa3b28c8957515f464026710e63a03
Retest Date:	May 13th, 2025
Final Report Date:	May 13th, 2025

5.2 Scope

The following smart contracts were in the scope of the security review:

Contracts
aave/AaveV3Strategy.sol
aave/AaveV3StrategyBase.sol
euler/EulerV2Strategy.sol
euler/EulerV2StrategyBase.sol
silo/SiloV2Strategy.sol
silo/SiloV2StrategyBase.sol

5.3 Findings Count

Severity	Count
Critical	
High	
Medium	
Low	01
Info	
Gas	

Severity	Count
Total	01

5.4 Findings Summary

Overall, the code is well-written. During the review, a total of One (1) issues were found. Out if which Zero (0) were Critical and High issues.

ID	Title	Severity	Status
L-01	Floating/Outdated pragma	Low	Fixed

6. Findings

[L-01] Floating/Outdated pragma

Description: All contracts across the codebase use the following pragma statement:

```
pragma solidity ^0.8.13;
```

Contracts should be deployed with the same compiler version and flags used during development and testing. An outdated pragma version might introduce bugs that affect the contract system negatively or recent compiler versions may have unknown security vulnerabilities.

Recommendation: It is recommended to lock the pragma to a latest and specific version of the compiler.

Team Response

> We should use a floating pragma in our contracts since the underlying dependencies rely on different pragma versions. That said, I agree the version used was a bit outdated. I've updated it to ^0.8.20, which should resolve the issue.