

Berachain

Smart Contract Security Assessment

VERSION 1.1



AUDIT DATES: August 20th to August 20th, 2025
AUDITED BY: said

Contents

1	Introduction	2
1.1	About Zenith	3
1.2	Disclaimer	3
1.3	Risk Classification	3
<hr/>		
2	Executive Summary	3
2.1	About Berachain	4
2.2	Scope	4
2.3	Audit Timeline	5
2.4	Issues Found	5
<hr/>		
3	Findings Summary	5
<hr/>		
4	Findings	6
4.1	Informational	7

1

Introduction

1.1 About Zenith

Zenith assembles auditors with proven track records: finding critical vulnerabilities in public audit competitions.

Our audits are carried out by a curated team of the industry's top-performing security researchers, selected for your specific codebase, security needs, and budget.

Learn more about us at <https://zenith.security>.

1.2 Disclaimer

This report reflects an analysis conducted within a defined scope and time frame, based on provided materials and documentation. It does not encompass all possible vulnerabilities and should not be considered exhaustive.

The review and accompanying report are presented on an "as-is" and "as-available" basis, without any express or implied warranties.

Furthermore, this report neither endorses any specific project or team nor assures the complete security of the project.

1.3 Risk Classification

SEVERITY LEVEL	IMPACT: HIGH	IMPACT: MEDIUM	IMPACT: LOW
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

2

Executive Summary

2.1 About Berachain

Berachain is a high-performance EVM-Identical Layer 1 blockchain utilizing Proof-of-Liquidity (PoL) and built on top of the modular EVM-focused consensus client framework BeaconKit.

2.2 Scope

The engagement involved a review of the following targets:

Target	morpho-blue
Repository	https://github.com/berachain/morpho-blue
Commit Hash	a94a5f1895b890a0108f8be2e4a9bbda9b9e7146
Files	Changes with PR-4

2.3 Audit Timeline

August 20, 2025	Audit start
August 20, 2025	Audit end
August 28, 2025	Report published

2.4 Issues Found

SEVERITY	COUNT
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	0
Informational	1
Total Issues	1

3

Findings Summary

ID	Description	Status
I-1	Unnecessary fee assignment and event emission when de-faultMarketFee is 0	Resolved

4

Findings

4.1 Informational

A total of 1 informational findings were identified.

[I-1] Unnecessary fee assignment and event emission when defaultMarketFee is 0

SEVERITY: Informational

IMPACT: Informational

STATUS: Resolved

LIKELIHOOD: Low

Target

- [Morpho.sol#L174-L178](#)

Description:

The Berachain fork of Morpho Blue introduces defaultMarketFee, which is assigned to newly created markets. However, it unnecessarily sets the fee and emits a SetFee event even when defaultMarketFee is 0.

Recommendations:

Consider skipping fee assignment and the SetFee event when defaultMarketFee is 0.

```
function createMarket(MarketParams memory marketParams) external {
    Id id = marketParams.id();
    require(isIrmEnabled[marketParams.irm], ErrorsLib.IRM_NOT_ENABLED);
    require(isLltvEnabled[marketParams.lltv], ErrorsLib.LLTV_NOT_ENABLED);
    require(market[id].lastUpdate == 0, ErrorsLib.MARKET_ALREADY_CREATED);

    // Safe "unchecked" cast.
    market[id].lastUpdate = uint128(block.timestamp);
    market[id].fee = uint128(defaultMarketFee);
    if (defaultMarketFee != 0) {
        (market[id].fee = uint128(defaultMarketFee);
        emit EventsLib.SetFee(id, defaultMarketFee);
    }
    idToMarketParams[id] = marketParams;
```

```
    emit EventsLib.CreateMarket(id, marketParams);  
    emit EventsLib.SetFee(id, defaultMarketFee);  
  
    // Call to initialize the IRM in case it is stateful.  
    if (marketParams.irm != address(0))  
        Iirm(marketParams.irm).borrowRate(marketParams, market[id]);  
}
```

Berachain: Resolved with [@59df023e10a...](#)

Zenith: Verified