Euler EVK
Security Review

Solo review by:
M4rio.eth, Security Researcher

January 23, 2025

Contents

1 Introduction

1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at cantina.xyz

1.2 Disclaimer

A security review is a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While the review endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that a security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

1.3 Risk assessment

Severity	Description
Critical	*Must* fix as soon as possible (if already deployed).
High	Leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.
Medium	Global losses <10% or losses to only a subset of users, but still unacceptable.
Low	Losses will be annoying but bearable. Applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.
Gas Optimization	Suggestions around gas saving practices.
Informational	Suggestions around best practices or readability.

1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

2 Security Review Summary

Euler Labs is a team of developers and quantitative analysts building DeFi applications for the future of finance.

On Jan 20th the security researcher conducted a review of euler-vault-kit on commit hash 3e5c2fea. **1** issue was identified:

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	0	0	0
Low Risk	0	0	0
Gas Optimizations	0	0	0
Informational	1	1	0
Total	**1**	**1**	**0**

3 Findings

3.1 Informational

3.1.1 Reentrancy check verification on the nested vaults

Severity: Informational

Context: Governance.sol#L281

Description: During the Cantina competition, two researchers reported a low-risk vulnerability in the EVK code (Issue 3.1.28 of the Cantina Competition Report). The team decided to fix this issue by adding a `getQuote` function call to the oracle router whenever the new liquidation LTV being set is greater than zero.

```
if (!newLiquidationLTV.isZero()) {
    // Ensure that this collateral can be priced by the configured oracle
    (, IPriceOracle _oracle, address _unitOfAccount) = ProxyUtils.metadata();
    _oracle.getQuote(1e18, collateral, _unitOfAccount);
}
```

This change caused issues for governors when dealing with pull-based oracles. To address this, the team has decided to revert the change, which will effectively reopen the bug reported in `Issue 3.1.28`.

Recommendation: While the bug itself is of low severity and we agree with simply documenting it properly, if the reentrancy check for the collateral is still desired, a more complex solution can be implemented in the `setLTV` function if new liquidation LTV being set is greater than zero:

```
IERC4626 stub4626 = IERC4626(collateral); // assume it's a valid ERC4626
uint256 codeSize;
assembly {
    codeSize := extcodesize(stub4626)
} // check the address has code, a collateral should have code tho so maybe this can be deleted
if (codeSize > 0) {
    try stub4626.asset() { // trying to see if the collateral is an ERC4626
        try IERC4626(stub4626.asset()).convertToAssets(1e18) { // trying to price one share, should revert with
        ↪  E_Reentrancy if it's this vault
        } catch Error(string memory reason) {
            if (keccak256(bytes(reason)) == keccak256(bytes("E_Reentrancy"))) { // revert if it's reentrancy
            ↪  error
                revert(reason);
            }
        } catch {
            // do nothing in any other case
        }
    } catch {
        // do nothing in any other case
    }
}
```

This solution is suggested only if the team wants to implement additional checks. However, in our opinion, the fix is not worth the complexity and should instead be addressed through proper documentation, which was already completed by the team in the reviewed commit.

Euler: Addressed by documenting the behavior.

M4rio.eth: Verified.