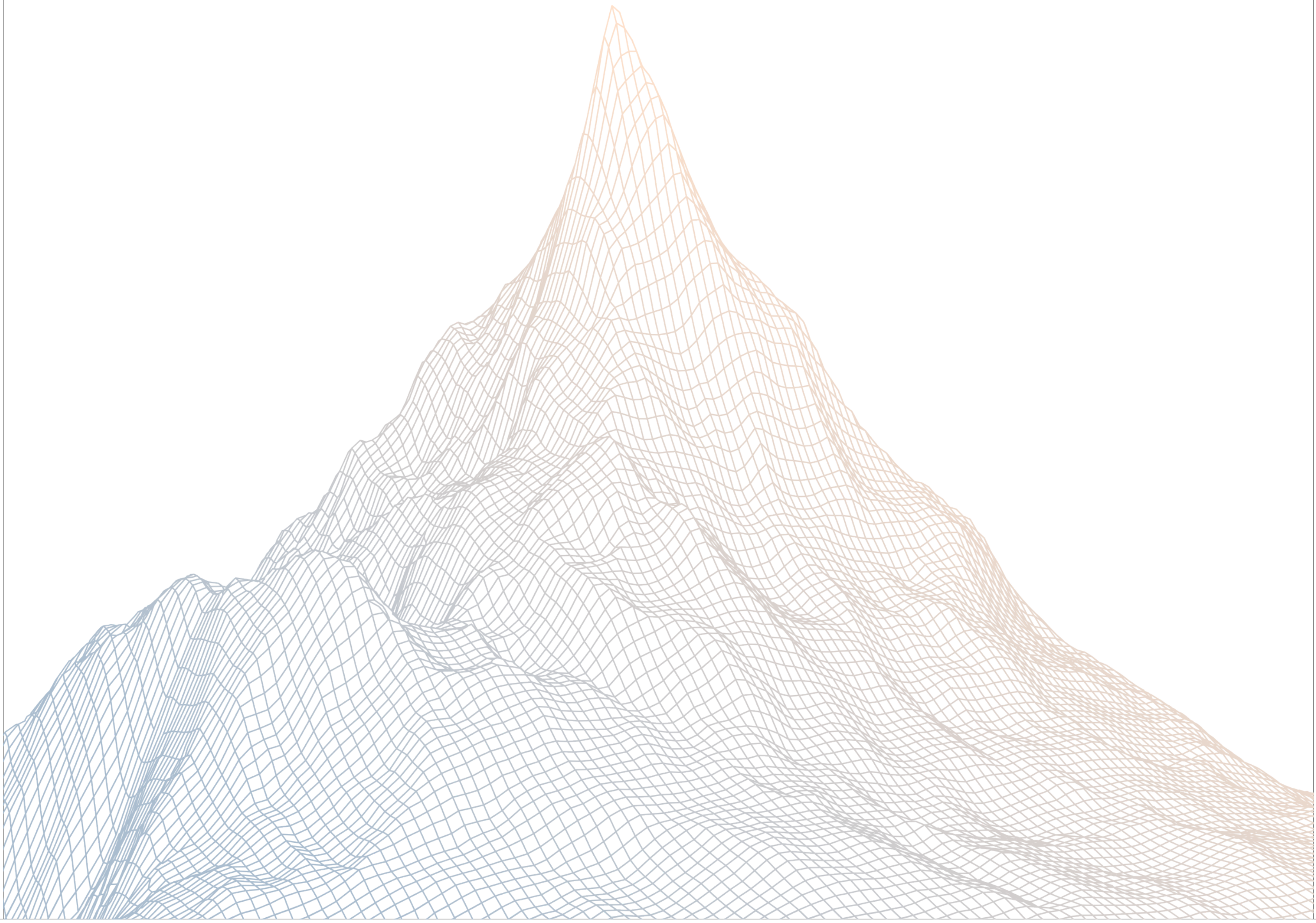
Zenith

Orderly

Smart Contract Security Assessment

VERSION 1.1



AUDIT DATES: April 16 to April 24, 2025

AUDITED BY: ethersky
ladboy

Contents

1 Introduction

1.1 About Zenith

Zenith is an offering by Code4rena that provides consultative audits from the very best security researchers in the space. We focus on crafting a tailored security team specifically for the needs of your codebase.

Learn more about us at https://code4rena.com/zenith.

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 Orderly

Orderly is a combination of an orderbook-based trading infrastructure and a robust liquidity layer offering perpetual futures orderbooks. Unlike traditional platforms, Orderly doesn't have a front end; instead, it operates at the core of the ecosystem, providing essential services to projects built on top of it.

With Orderly, anyone can create a trading application thanks to our seamless plug-and-play experience leveraging our liquidity and composability.

2.2 Scope

The engagement involved a review of the following targets:

Target	evm-cross-chain
Repository	https://github.com/OrderlyNetwork/evm-cross-chain
Commit Hash	`313a7892cbd03a665da69a1b61032a4c1865e085`
Files	`contracts/* (excluding test/mock files)`

Target	evm-cross-chain mitigation review
Repository	https://github.com/OrderlyNetwork/evm-cross-chain
Commit Hash	`2234b02a2d7149bd07636b2af0593abc0df2f793`
Files	`contracts/* (excluding test/mock files)`

Target	cross-chain-v2
Repository	https://github.com/OrderlyNetwork/cross-chain-v2
Commit Hash	`86cfc1ebfb37dab801955f88e946f7bf4359238e`
Files	`contracts/* (excluding test/mock files)`

Target	cross-chain-v2 mitigation review
Repository	https://github.com/OrderlyNetwork/cross-chain-v2
Commit Hash	`462d62cdc0776a66db17ac10419280e16d5e2746`
Files	`contracts/* (excluding test/mock files)`

2.3 Audit Timeline

April 16, 2025	Audit start
April 24, 2025	Audit end
April 24, 2025	Report published

2.4 Issues Found

SEVERITY	COUNT
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	3
Informational	2
Total Issues	**5**

3

Findings Summary

ID	Description	Status
L-1	The onlyProxy modifier is missing in the upgradeTo function	Resolved
L-2	Consider use safeTransfer to rescue ERC20 token from the CrossChainRelayV2.sol	Resolved
L-3	The decimal value of tokens must be defined before performing any operations	Acknowledged
I-1	Consider emit event properly in state-change functions in CrossChainRelayerV2.sol	Resolved
I-2	Admin can pause the CrossChainRelayerV2.sol but has no impact in message relaying	Resolved

4

Findings

4.1 Low Risk

A total of 3 low risk findings were identified.

[L-1] The onlyProxy modifier is missing in the upgradeTo function

SEVERITY: Low	IMPACT: Low
STATUS: Resolved	LIKELIHOOD: Low

Target

- VaultCrossChainManagerUpgradeable.sol

Description:

The `VaultCrossChainManagerUpgradeable` and `LedgerCrossChainManagerUpgradeable` are `UUPSUpgradeable`, and they have the `upgradeTo` function. This function is intended to be called within `proxy` contracts rather than implementation contracts. However, these functions lack the `onlyProxy` modifier.

- VaultCrossChainManagerUpgradeable.sol#L88

```
/// @notice Upgrades the implementation contract
/// @dev Only callable by owner through proxy
/// @param newImplementation Address of new implementation contract
function upgradeTo(address newImplementation) public override onlyOwner {
    _upgradeToAndCallUUPS(newImplementation, new bytes(0), false);
}
```

Recommendations:

```
function upgradeTo(address newImplementation) public override onlyOwner {
function upgradeTo(address newImplementation) public override onlyOwner
    onlyProxy {
    _upgradeToAndCallUUPS(newImplementation, new bytes(0), false);
}
```

Orderly: Resolved with @1c544156ef...

Zenith: Verified.

[L-2] Consider use `safeTransfer` to rescue ERC20 token from the `CrossChainRelayV2.sol`

SEVERITY: Low	IMPACT: Low
STATUS: Resolved	LIKELIHOOD: Low

Target

- CrossChainRelayV2.sol

Description:

This function is designed to rescue ERC20 token from the smart contract.

```
/// @notice Withdraws ERC20 tokens from the contract
/// @param token Token address
/// @param to Recipient address
/// @param amount Amount of tokens to withdraw
function withdrawToken(address token, address to, uint256 amount)
    external onlyOwner {
    IERC20(token).transfer(to, amount);
}
```

However, certain token such as USDT does not return bool when transferring function is triggered,

then `withdrawToken` will revert.

Recommendations:

Use `safeTransfer` to handle such token

- Docs Openzeppelin - SafeERC20

Orderly: Resolved with @6a2a1af020...

Zenith: Verified.

[L-3] The decimal value of tokens must be defined before performing any operations

SEVERITY: Low	IMPACT: Low
STATUS: Acknowledged	LIKELIHOOD: Low

Target

- LedgerCrossChainManagerUpgradeable.sol

Description:

Tokens are converted to match the `decimal` of the `target chain` between the `ledger` and the `vault`.

- LedgerCrossChainManagerUpgradeable.sol#L384-L385

```
function burn(RebalanceTypes.RebalanceBurnCCData memory burnData)
    external override onlyLedger {
    // Convert token amount to destination chain decimals
    uint128 cvtTokenAmount =
        convertDecimal(burnData.amount, burnData.tokenHash, chainId,
    burnData.burnChainId);
    burnData.amount = cvtTokenAmount;

    bytes memory payload = abi.encode(burnData);

    _sendMessage(message, payload);
}
```

The `token decimals` for each `chain` are defined by the `owner`.

- LedgerCrossChainManagerUpgradeable.sol#L237

```
function setTokenDecimal(bytes32 tokenHash, uint256 tokenChainId,
    uint128 decimal) external onlyOwner {
    _setTokenDecimal(tokenHash, tokenChainId, decimal);
}
```

If a `token's decimal` is not set, it defaults to `0`, but this default value should not be used in any operations. However, the `convertDecimal` function still operates normally, even if the `decimal` is unset.

- LedgerCrossChainManagerUpgradeable.sol#L119-L127

```
function convertDecimal(uint128 tokenAmount, bytes32 tokenHash,
    uint256 srcChainId, uint256 dstChainId)
    public
    view
    returns (uint128)
{
    uint128 srcDecimal = getTokenDecimal(tokenHash, srcChainId);
    uint128 dstDecimal = getTokenDecimal(tokenHash, dstChainId);
    return convertDecimal(tokenAmount, srcDecimal, dstDecimal);
}
```

Recommendations:

```
function convertDecimal(uint128 tokenAmount, bytes32 tokenHash,
    uint256 srcChainId, uint256 dstChainId)
    public
    view
    returns (uint128)
{
    uint128 srcDecimal = getTokenDecimal(tokenHash, srcChainId);
    uint128 dstDecimal = getTokenDecimal(tokenHash, dstChainId);

    require(srcDecimal ≠ 0, decimal not set);
    require(dstDecimal ≠ 0, decimal not set);

    return convertDecimal(tokenAmount, srcDecimal, dstDecimal);
}
```

Orderly: Acknowledged

4.2 Informational

A total of 2 informational findings were identified.

[I-1] Consider emit event properly in state-change functions in `CrossChainRelayerV2.sol`

SEVERITY: Informational	IMPACT: Informational
STATUS: Resolved	LIKELIHOOD: Low

Target

- CrossChainRelayV2.sol

Description:

Consider emit events in a list of state-change functions below in `CrossChainRelayerV2.sol`

- `function addChainIdMapping(uint256 _chainId, uint32 _eid) external onlyOwner`
- `function setCCManager(address _ccManager) external onlyOwner`
- `function setMethodOption(uint8 _method, uint128 _lzGas, uint128 _lzValue) external onlyOwner`

Recommendations:

Emit event properly in state-change functions above.

Orderly: Resolved with @52a3c2c7c1f...

Zenith: Verified.

[I-2] Admin can pause the `CrossChainRelayerV2.sol` but has no impact in message relaying

SEVERITY: Informational	IMPACT: Informational
STATUS: Resolved	LIKELIHOOD: Low

Target

- CrossChainRelayV2.sol

Description:

`CrossChainRelayV2` inherit from `OApp`

```
contract CrossChainRelayV2 is IOrderlyCrossChain, OApp,
    CrossChainRelayDataLayoutV2 {
```

and `OApp` inherit from `OAppUpgradeable.sol`

```
abstract contract OApp is OAppUpgradeable, OAppOptionsType3Upgradeable {
```

The owner of the Oapp can pause the smart contract.

```
function pause() public onlyOwner {
    _pause();
}

function unpause() public onlyOwner {
    _unpause();
}
```

However, the message continue to be relayed and executed even when the `OApp` is paused by owner.

Recommendations:

Check if the OApp is paused before executing the message.

Orderly: Resolved with @213d6f2e8...

Zenith: Verified. When the smart contract is paused, both `send` and `receive` revert.