# Immutable Internal Contract Audit

# Immutable Signed Zone v2

| | |
|---|---|
| **Internal Auditors** | Peter Robinson |
| **Assessment Date** | May 1, 2024 |
| **Final Report** | May 2, 2024 |

Previous audits:

- August 2023: Audit information for v1: https://github.com/immutable/contracts/blob/main/audits/trading/202308-audit-information-seaport.md

# Contents

# Description

Immutable operates a global off-chain orderbook across Immutable zkEVM chains and relies on the Seaport protocol for on-chain settlement. The orderbook primarily intends to:

- Centralise liquidity
- Enforce payment of fees (protocol, royalty, marketplace)

The Immutable Signed Zone (v2) is an implementation of SIP-7: Server-Signed Orders. The reasons Immutable has chosen this solution is:

- It requires orders to be known by the orderbook prior to fulfillment
- It allows the orderbook to refuse fulfillment of an order at its discretion (e.g. for gasless cancellations, or compromised collections)
- It allows for arbitrary logic to be defined off-chain and enforced on-chain (e.g. to enforce fees)

# Scope

Commit: `f6dffe08db1ff2daead3abe0c22b44d792ce5e59`

Fix Review Commit: 18fe47b33cce1c511c3e9c648807589f48850c56

| Asset | Description |
|---|---|
| Smart Contracts | ImmutableSignedZoneV2.sol<br><br>Which includes from Immutable:<br>● ZoneAccessControl.sol<br>● ZoneAccessControlEventsAndErrors.sol<br>● SIP5EventsAndErrors.sol<br>● SIP5Interface<br>● SIP6EventsAndErrors<br>● SIP6Interface<br>● SIP7EventsAndErrors<br>● SIP7Interface<br><br>Which includes from Sea Port (v1.6):<br>● ZoneInterface.sol which imports:<br>○ ConsiderationStructs.sol<br>○ ConsiderationEnums.sol<br>○ PointerLibraries.sol<br>● ConsiderationStructs.sol<br><br>Which includes from Open Zeppelin (v5.0.2):<br>● AccessControlEnumerable.sol which imports:<br>○ AccessControl.sol<br>○ IAccessControl.sol<br>○ IAccessControlEnumerable.sol<br>○ EnumerableSet.sol<br>○ Context.sol |

| | |
|---|---|
| | ○ Strings.sol<br>○ Math.sol<br>○ SignedMath.sol<br>○ ERC165.sol<br>○ ERC2981.sol<br>○ IERC2981.sol<br>● ECDSA.sol<br>● MessageHashUtils.sol<br>● Math.sol<br>● ERC165.sol |
| Threat model | https://github.com/immutable/contracts/blob/main/audits/trading/202404-threat-model-immutable-signed-zone-v2.md |

# Team’s Greatest Concerns

- The team would like a general review. They have no specific concerns.

# Basic Contract Analysis

The table below analyzes[1] the contracts.

| Type | File | Lines | nSLOC | Complexity |
|---|---|---|---|---|
| Contract | ImmutableSignedZoneV2 | 613 | 243 | 195 |
| Abstract contract | ZoneAccessControl | 51 | 24 | 23 |

[1] Analysis produced using the Solidity Metrics VSCode extension.

# Smart Contracts

## ImmutableSignedZoneV2

Roles:

- DEFAULT_ADMIN_ROLE: Role administrator for ZONE_MANAGER_ROLE
- ZONE_MANAGER_ROLE: Administrators that configure the zone.

External or Public Functions[2] that modify state:

| Function Name | Function Selector | Authorisation Check |
|---|---|---|
| addSigner | eb12d61e | Only ZONE_MANAGER_ROLE |
| grantRole | 2f2ff15d | Msg.sender must be the role admin for the role |
| removeSigner | 0e316ab7 | Only ZONE_MANAGER_ROLE |
| renounceRole | 36568abe | Msg.sender can only remove their own access. |
| revokeRole | d547741f | Msg.sender must be role admin for the role being revoked. For DEFAULT_ADMIN_ROLE, they must be another account with DEFAULT_ADMIN_ROLE. |
| updateAPIEndpoint | 297234d7 | Only ZONE_MANAGER_ROLE |
| updateDocumentationURI | 0a904f08 | Only ZONE_MANAGER_ROLE |

External or Public Functions that do not modify state:

| Function Name | Function Selector | Notes |
|---|---|---|
| DEFAULT_ADMIN_ROLE | a217fddf | |
| ZONE_MANAGER_ROLE | c6e95ae7 | |
| getRoleAdmin | 248a9ca3 | |
| getRoleMember | 9010d07c | |
| getRoleMemberCount | ca15c873 | |
| getSeaportMetadata | 2e778efc | |
| hasRole | 91d14854 | |
| sip7Information | d600940e | |
| supportsInterface | 01ffc9a7 | |

[2] The list of functions was determined using `forge inspect ImmutableSignedZoneV2 methods`

|  |  |  |
|---|---|---|
| validateOrder | 17b1f942 |  |

Upgradeable checks: This contract is not upgradeable.

Analysis of code logic: See Abused Case Findings below

# Abuser Test Cases and Findings

## No Substandards Supported

Classification: Critical

Description: The _validateSubstandards function could be passed a zero length context. This would correspond to no substandards being supported. This should revert.

Action: Revert if no sub-standards are supported.

Status: Team will fix as per action.

Fix review: Completed.

## Solidity Version

Classification: Informational

Description: File currently have the Solidity version pragma:

```
pragma solidity ^0.8.20;
```

A developer could compile the code using Solidity version 0.8.25 or later. The resulting bytecode may have Cancun hardfork specific features included in the code. Unexpected results could occur when running on Immutable zkEVM as Immutable zkEVM does not support Cancun hardfork features (as at May 1, 2024).

Action: Specify the precise Solidity version as 0.8.20.

Status: Team will fix as per action.

Fix review: completed.

## Upgradeability not documented

Classification: Documentation

Description: ImmutableSignedZoneV2 does not document the approach to upgrade. Developers and deployers could become confused as to how to upgrade the contract.

Action: Document the approach to upgrade. This could be to say that the contract is not upgradeable, and that if a bug is found, that a new contract should be deployed and contracts and off-chain infrastructure using this contract should be pointed to the address of the newly deployed contract.

Status: Team has documented upgrade process in the contract level comment.

Fix review: completed.

# Difference Between V1 and V2 not documented

Classification: Documentation

Description: The README.md file for Immutable Signed Zone V2 should explain the difference between V1 and V2. This will allow consumers of the code to understand whether they should use V1 or V2, and if they are using V1, whether they should upgrade or not.

Action: Document the differences between V1 and V2 in the README.md file.

Status: Team will document the differences.

Fix review: completed.

# Check - Effects - Interactions

Classification: Informational

Description: The Check - Effects - Interactions pattern says that checks should be done, then affects updating state, and then finally cross-contract interactions. This helps to prevent reentrancy style bugs.

In ZoneAccessControl, the revokeRole and renounceRole functions should have the checking to ensure this last DEFAULT_ADMIN_ROLE isn't being removed before the call to super.revokeRole or super.renounceRole.

Action: Switch the order to Check - Effects - Interactions.

Status: Team will update the code.

Fix review: Completed.

# Gas Usage

Classification: Informational

Description: The SignerInfo struct contains two booleans. This state information could be contained in a single enum or integer: 0 = not used; 1 = active; 2 = previously active. Doing this would reduce the number of sstore operations.

In _validateSubstandards function, there is the following code:

```
        _validateSubstandard3(context[startIndex:],
```

This could just be the following, which I believe would save some gas.

```
        _validateSubstandard3(context,
```

Action: Consider and evaluate improvements.

Status: Team has acknowledged the gas usage and will defer implementation of this to a later date.