1inch limit order protocol audit

16 Oct 2021, Igor Gulamov

Introduction

Igor Gulamov conducted the audit of 1inch limit order protocol smart contracts.

This review was performed by an independent reviewer under fixed rate.

Scope

Smart contracts from 1inch/limit-order-protocol.

Issues

We find no major and no critical issues. We consider commit 9d118307df7acc3bcef73407f3964acd6aa0f35c as a safe version from the informational security point of view.

Warnings

Comments

ChainlinkCalculator.sol#L25-L29

We recommend to precompute constant multipliers here as

```
uint256 private constant _ORACLE_DENOMINATOR = 1e18;
uint256 private constant _INVERSE_DENOMINATOR =
_ORACLE_DENOMINATOR/_SPREAD_DENOMINATOR;
uint256 private constant _DIRECT_DENOMINATOR =
_ORACLE_DENOMINATOR*_SPREAD_DENOMINATOR;
```

This is valid if `_SPREAD_DENOMINATOR` is multiplier of `_ORACLE_DENOMINATOR`.

ERC721Proxy.sol#L15

ERC721ProxySafe.sol#L15

ERC1155Proxy.sol#L15

Unnecessary type conversion.

PredicateHelper.sol#L49-L65

Arguments and operands order are not corresponding. Ensure, that function is used right way everywhere. Also, we recommend to rewrite arguments or operands order.

Severity Terms

Comment

Comment issues are generally subjective in nature, or potentially deal with topics like "best practices" or "readability". Comment issues in general will not indicate an actual problem or bug in code.

The maintainers should use their own judgment as to whether addressing these issues improves the codebase.

Warning

Warning issues are generally objective in nature but do not represent actual bugs or security problems.

These issues should be addressed unless there is a clear reason not to.

Major

Major issues will be things like bugs or security vulnerabilities. These issues may not be directly exploitable, or may require a certain condition to arise in order to be exploited.

Left unaddressed these issues are highly likely to cause problems with the operation of the contract or lead to a situation which allows the system to be exploited in some way.

Critical

Critical issues are directly exploitable bugs or security vulnerabilities.

Left unaddressed these issues are highly likely or guaranteed to cause major problems or potentially a full failure in the operations of the contract.