yAudit Origami oracles adapters Review

Review Resources:

- Repository

Auditors:

- Adriro
- Panda

Table of Contents

Review Summary

Origami oracles adapters

Origami oracles adapters provide a set of oracles for different liquid restaked ETH.

The contracts of the Origami oracles adapters Repo were reviewed over two days. The code review was performed by two auditors between 27 June and 28 June, 2024. The repository was under active development during the review, but the review was limited to the latest commit at the start of the review. This was commit b2f684164f9a48cf73ae7c1e9e29ad743fa85301 for the Origami oracles adapters repo.

Scope

The scope of the review consisted of the following contracts at the specific commit:

```
src/exchange-rate-adapters/
├── EzEthToEthExchangeRateChainlinkAdapter.sol
├── RsEthToEthExchangeRateChainlinkAdapter.sol
├── RswEthToEthExchangeRateChainlinkAdapter.sol
├── SwEthToEthExchangeRateChainlinkAdapter.sol
├── WeEthToEthExchangeRateChainlinkAdapter.sol
```

This review is a code review to identify potential vulnerabilities in the code. The reviewers did not investigate security practices or operational security and assumed that privileged accounts could be trusted. The reviewers did not evaluate the security of the code relative to a standard or specification. The review may not have identified all potential attack vectors or areas of vulnerability.

yAudit and the auditors make no warranties regarding the security of the code and do not warrant that the code is free from defects. yAudit and the auditors do not represent nor imply to third parties that the code has been audited nor that the code is free from defects. By deploying or using the code, Origami oracles adapters and users of the contracts agree to use the code at their own risk.

Oracle analysis

In this section, we focused on analyzing the different oracles used by the contracts. This included evaluating their reliability and effectiveness in providing accurate data for the contract operations.

Overview

	ezETH	rsETH	swETH	rswETH	weETH
Protocol	Renzo Protocol	KelpDAO	Swell Network	Swell Network	Ether.fi
Assets	wBETH, stETH	stETH, ETHx, sfrxETH, Native ETH	ETH	ETH	ETH
Oracle address	Etherscan	Etherscan	Etherscan	Etherscan	Etherscan
Atomic price manipulation	Possible	Possible	No	No	No
Hardcoded oracle values	None	stETH and frxETH values assumed to be 1:1 with ETH	None	None	None
Oracle price	1.013	1.014	1.063	1.011	1.042
Market price	1.006	1.007	1.060	0.998	1.040
Chainlink usage	Yes	No	No	No	No
EOA controls the price	No	No	Yes	Yes	Yes (limited)

	ezETH	rsETH	swETH	rswETH	weETH
Upgradable contracts	Yes	Yes	Yes	Yes	Yes
Withdrawals enabled	Limited	Yes	Yes	No	Yes

Note: market and oracles prices were taken on 27th June, 2024.

Renzo Oracle

Protocol: Renzo Protocol

Assets: wBETH, stETH

Oracle Address: Etherscan

Atomic price manipulation

The current implementation of the price determination mechanism in the Renzo Protocol is subjective to potential manipulation. The price is determined by the restake manager, relying on the number of operators' respective balances. We have identified a potential attack vector that could be exploited to manipulate the price of the token. The `withdrawalQueue` balance is checked to assess the protocol price. A transfer of tokens to the queue will increase the price. This mechanism could be exploited to artificially inflate the price of the token.

Chainlink usage

The Renzo Protocol uses Chainlink oracles to determine the price of the `stETH` token. This price doesn't reflect the real value of an stETH if it was to be withdrawn. However the `stETH` has the biggest liquidity of all the existing LSTs and is less likely to have a huge price movement.

Withdrawal

While withdrawals are enabled, the amount of funds available to withdraw are really limited: an amount in the order of one ETH has been observed to be withdrawable. In a future update, the protocol will introduce a withdrawal queue mechanism allowing users to request a withdrawal regardless of the buffer amount.

KelpDAO Oracle

Protocol: rsETH Protocol

Assets: stETH, ETHx, sfrxETH, Native ETH

Oracle Address: Etherscan

Hardcoded price values

The stETH price is hardcoded to `10**18`. While this is valid under normal circumstances, in case of a price change due for example to slashing the price will drop in the market but won't be reflected immediately in the contract.

The SfrxETHPriceOracle contract used to price sfrxETH internally calls `sfrxETH::pricePerShare()` returning the conversion from sfrxETH to frxETH, hardcoding a 1:1 relation between frxETH and ETH.

Atomic Price Manipulation

A transfer of tokens to the LRTDepositPool can manipulate the price. This vulnerability could be exploited to artificially inflate the token price. The attack is possible via a `balanceOf()` call.

Withdrawal

Unstake requests are processed in 7-10 days, subject to exit queue on Ethereum network and delays imposed by EigenLayer.

Swell Network Oracle

Protocol: Swell Network

Assets: ETH

Oracle Address: Etherscan

Centralization risk

The price determination mechanism in the swETH and rswETH protocols relies on an off-chain entity to calculate the price. This is done through a protected call to the `reprice()` function, based on the input data provided by the repricer. The repricer is a single address entity, not a consensus of multiple entities, which carry a risk of centralization.

Withdrawal

Withdrawal is enabled for swETH but not for rswETH. The delay is on average 12 days.

weETH Oracle

Protocol: ether.fi Protocol

Assets: eETH (ETH)

Oracle Address: Etherscan

weETH is the wrapped variant of eETH, an LST with rebasing mechanics.

Centralization risk

Updates to the `rebase()` function that controls the total amount of ETH owned by the protocol are handled through the EtherFiAdmin contract. Although the owner is a timelock, the contract implements a whitelist of admins allowed to control functionality. One of these current admins is an EOA. However, accrued rewards have on-chain requirements that limit the rebasing amount.

Withdrawal

eETH redemption process can take up to 14+ days as noted by their interface.

> EigenLayer pods can take 7+ days to exit and normal validator exits can take 7+ days for a total of 14+ days before your funds can be claimed.

Findings Explanation

Findings are broken down into sections by their respective impact:

- Critical, High, Medium, Low impact
  - These are findings that range from attacks that may cause loss of funds, impact control/ownership of the contracts, or cause any unintended consequences/actions that are outside the scope of the requirements.
- Gas savings
  - Findings that can improve the gas efficiency of the contracts.
- Informational
  - Findings including recommendations and best practices.

Critical Findings

None

High Findings

None

Medium Findings

None

Low Findings

None

Gas Saving Findings

1. Gas - Cache oracle address in ezETH and rsETH

Both of these adapters execute an initial reference lookup in each query.

Technical Details

The implementation of EzEthToEthExchangeRateAdapter.sol fetches the oracle reference from the restake manager.

```
        uint256 rate =
IRenzoOracle(RENZO_RESTAKE_MANAGER.renzoOracle()).calculateRedeemAmount(
            1 ether, EZ_ETH.totalSupply(), _currentValueInProtocol
        );
```

Similarly, RsEthToEthExchangeRateAdapter.sol queries the LRTConfig contract to fetch the address of the LRTOracle.

```
        IKelpLRTOracle lrtOracle =
IKelpLRTOracle(KELP_LRT_CONFIG.getContract(LRT_ORACLE));
```

Impact
Gas savings.

Recommendation
Provide these addresses as constants.

Developer Response
I intentionally followed the underlying protocol implementations where they do not mark these addresses as constants. That implied to me that the addresses may change and shouldn't be marked as constants.

Informational Findings

1. Informational - Make sure the origami oracle address can be updated
The Origami Oracles are built on upgradable contracts. While we do not believe it's necessary to make the oracles themselves upgradable, we advise ensuring that the oracle address can be updated in any contract utilizing it.

Technical Details
All protocols used by the oracles are based on upgradable contracts. Although we do not anticipate changes to the interface, it is prudent to be prepared for updates.

Impact
Informational

Recommendation
Make sure you can update the contract address if the oracle needs to be updated.

Developer Response
This oracle adapter is intended to be used to price debt<>collateral within Morpho Blue markets, where the Oracle is immutable – in the event of a required update, a new Morpho Blue market shall be created, rather than a contract update/migration.

2. Informational - Market price and exchange rate tradeoff
Critical events could create a significant divergence between market prices and exchange rates provided by protocols.

Technical Details

In the unlikely event of a severe scenario, such as a hack or protocol malfunction, that could result in loss of funds, asset locks, or excessive withdrawal delays, a big discrepancy could be potentially observed between the oracles that follow market prices and the oracles that rely on exchange rates, which are mostly determined by the ratio of protocol-owned assets to total minted tokens.

An efficient market could quickly react to these scenarios. However, it is unclear if and when protocols would factor such events into their exchange rates.

Among the various adapters, the Swell implementation clearly demonstrates the potential issue. Its exchange rate is essentially determined by what protocol administrators feed into the `reprice()` function. Assets owned by the protocol are effectively measured by the sum of arguments supplied to this function.

```
    uint256 totalReserves = _preRewardETHReserves + _newETHRewards;
```

It is difficult to determine or estimate how quickly these updates could be reflected in the contract’s state. As long as changes are not recorded on-chain, the exchange rate will remain unchanged.

Additionally, it’s important to note that these protocols carry risks from associated dependencies. For example, Kelp DAO’s architecture relies on three different LSTs (stETH, ETHx, sfrxETH), plus the integration with EigenLayer. As previously detailed, stETH and frxETH are assumed to have a 1:1 relationship with ETH, discounting the possibility of a depeg risk. If such an event were to occur, it wouldn’t be reflected in their oracle price.

Impact

Informational.

Recommendation

This ultimately represents a design decision that entails a tradeoff between the risks of one approach over the other.

Developer Response

Noted that there is a balance of risks between using a market price (where that market price may be manipulated) vs the exchange rate (which has centralisation risks and privy to the timing of updates).

Final remarks

The review highlighted operational details and risks such as atomic price manipulation and centralization. No critical vulnerabilities were found, with recommendations for gas efficiency optimization and awareness of potential market-exchange rate discrepancies.