SOLIDIFIED

Audit Report for Aztec - July 5, 2022

Summary

Audit Report prepared by Solidified covering the Aztec protocol Ethereum Bridge contract for Curve Bridge.

The following report covers the **Curve Bridge**.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief on 4 July 2022.

Audited Files

The source code has been supplied in the form of one public Github repository.

https://github.com/aztecProtocol/aztec-connect-bridges/

Commit Hash: `3b5edf619ef4d316d4430988b99f6cb0aac6a2b2`

```
src
|-- bridges
|   -- curve
       |-- CurveStEthBridge.sol
```

Intended Behavior

Smart contract responsible for depositing, managing and redeeming Defi interactions with the Curve protocol.

Code Complexity and Test Coverage

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of a smart contract system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**.

Note, that high complexity or lower test coverage does equate to a higher risk. Certain bugs are more easily detected in unit testing than in a security audit and vice versa. It is, therefore, more likely that undetected issues remain if the test coverage is low or non-existent.

Criteria	Status	Comment
Code complexity	Low	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	High	-

Issues Found

Issue #	Description	Severity	Status
1.	No minimum amount for Curve exchange	Minor	Resolved
2.	Compare Curve output amount with ERC20.balanceOf	Note	-

SOLIDIFIED

Audit Report for Aztec - July 5, 2022

Critical Issues

No issues found

Major Issues

No issues found

Minor Issues

1. No minimum amount for Curve exchange

The interaction with `CURVE_POOL.exchange` doesn't define a minimum output amount. This could lead to potential sandwich attacks.

See: https://ethereum.org/en/developers/docs/mev/#mev-examples-sandwich-trading

Recommendation

Pass a minimum amount as an additional parameter from off-chain or define a minimum amount as a percentage of the input amount to ensure at least a certain price range.

Informational Notes

2. Compare Curve output amount with ERC20.balanceOf

The `dy` variable after the `exchange` in the `_wrapETH` function could be compared with the actual `stETH.balanceOf(this).`

The same pattern as in `_unwrapETH` function with the ETH balance.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Aztec Protocol or its products. This audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors from legal and financial liability.

Oak Security GmbH