Stake2Care audit Report

This audit report provides a comprehensive analysis of the Stake2Care protocol, conducted by HHK, focusing on identifying potential security vulnerabilities, gas optimizations and assessing overall code quality.

Table of Contents

Project Presentation

Stake2Care

Stake2Care provides the ImpactVault smart contract that is an ERC 4626 Vault, which is used to donate the revenues of a value-accruing token to a NGO. In the present case, users will deposit Lido ST-ETH and the staking yield will be distributed to Doctors Without Borders (MSF).

On Deposit of stETH, the user receives MSF-ETH. Withdrawals are instantaneous and handled in stETH only.

Deposits into the ImpactVault are further mediated by the LidoImpactVaultDepositor to convert-and-deposit ETH as well. When converting ETH, the user can choose which proportion of the obtained stETH is to be deposited into the Impact Vault while the remainder is sent back to them.

Auditor presentation

HHK is a freelance smart contract developer and security researcher with a strong track record. He Often rank in the top 10 during audit contests and has conducted over 12 audits with yAudit. Specializing in smart contract security since early 2023, HHK's audit portfolio is available here. For inquiries, please contact hhk.contact@proton.me.

Scope

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

```
src/
├── ImpactVault.sol
├── ImpactVaultDepositor.sol
└── LidoImpactVaultDepositor.sol
```

The contracts of the Stake2Care Repo were reviewed over 1.5 days. 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 3823ab29145f9d1db9353bc6025261f745f471f9 for the Stake2Care repo.

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. By deploying or using the code, Stake2Care and users of the contracts agree to use the code at their own risk.

Findings Severity

Impact/Likelyhood	High	Medium	Low
High	Critical	High	Medium
Medium	High	Medium	Low
Low	Medium	Low	Informational

Suggestions to enhance performance (gas consumption) and architecture are provided as gas and informational findings.

Note that the assessment of findings severity is subjective and may vary depending on individual perspectives, system configurations, and specific use cases. What one user considers a critical issue may be deemed less severe by another based on different criteria and priorities.

Findings

Low Findings

1. Low - `ImpactVault` doesn't take `steth` potential precision loss into account

Technical Details

The `ImpactVault` assumes that the whole amount is transferred even though the Lido documentation says that in some case because of precision loss, few `wei` might not be transferred to the receiver.

This can result in the contract minting few extra `wei` of shares towards the depositor at the expense of previous depositors.
For example if a user deposits 1 `wei` the contract mints him back 1 `wei` of shares but received 0 `wei` of `steth`.

Additionally if the first deposit was to be very small (few `wei`) it would grow the share ratio by a lot and delay donations.

For example if one was to currently deposit 5 `wei` of `steth` as first depositor, only 4 `wei` would be sent to the `ImpactVault` while 5 shares would be minted.

Future deposits would be impacted as he functions `_convertToSharesCompute()` and `_convertToAssetsCompute` would mint using the share ratio instead of the 1-1 ratio.

If a depositor came after and deposited 1 `steth` he would receive 1.2e18 shares and as explained in info#1 his yield will not be used for donations until he and the whole vault breaks even (20%, which could take years).

Impact

Low.

Recommendation

While the auditor wasn't able to find a significant impact for this issue, it is recommended to ensure the amount of shares minted is always what's expected.

Consider checking the `steth.balanceOf()` before and after the transfer or use `steth.transferShares()` to avoids possible rounding issues.

Additionally consider adding a minimum deposit to make sure any `wei` loss is quickly repaid by the yield.

Developer Response

Developer:

Partially fixed.

- Added minimum deposit in a4a08127c89a3f0fe98dfdeeb81a2123c916ac2c.

- We decided to leave as is as further mitigations have a far higher gas cost and economic impact than the the 1 or 2 wei imprecision. Under a low 6Gwei gas price, an additional SLOAD operation would e.g. add ~ 12k Gwei overhead on deposit operations.

Auditor:

Consider using custom errors instead of `require` if gas is an important concern.

Developer:

Fixed in 9a1ee20541ae0a7655008b191f835f531fc16dc3.

Gas Saving Findings

1. Gas - Use unchecked math when it's safe

Technical Details

In the `collectDonations()` function multiple operations are safe and will not underflow/overflow.

- Line 274
- Line 277
- Line 287

Impact

Gas.

Recommendation

Consider using unchecked blocks.

Developer Response

Fixed in a4a08127c89a3f0fe98dfdeeb81a2123c916ac2c.

Informational Findings

1. Informational - Depositing after an `steth` slashing will not distribute depositor's yield until the whole vault debt resets

Technical Details

The functions `_convertToSharesCompute()` and `_convertToAssetsCompute` are called when depositing and exiting from the `ImpactVault`.

They are in charge of determining how much assets or shares should be received by the user. In most case they will assume a share ratio of 1.

But there is an exception if Lido validators get slashed and the `steth` ratio decreases, in this case the `totalShares` and `totalAsset` will be used to determine the share ratio. This ensures that new

depositors will not be impacted by the vaults debt as well that no bankrun happen as all users will be impacted the same.

The issue is that because new depositors will be minted shares at a higher ratio while the debt hasn't been recovered then their yield will not be donated until the debt of the whole vault has been erased.

Take this example:

- The vault has 100 `steth` and 100 shares, a slashing happens and `totalAsset` gets reduced by 2% and becomes 98 `steth`.
- A new user deposits 10 `steth`, he will receive roughly 10.20 shares.
- The user can withdraw anytime for at least 10 `steth` and then if the vault slowly clears out it's debt, the user will be able to withdraw his deposit + yield. For example when the vault is back at 1 share ratio the user can withdraw 10.2 `steth`.

Impact

Informational.

Recommendation

Consider allowing the yield of new users to be used to reimburse the vault debt or for donations instead of giving it to the new users until the vault debt is erased.

Developer Response

Acknowledged - it is a design choice as we decided against such an adjustment because:

- Doing so would imply keeping track of each user's unique `high watermark' level, which would significantly increase complexity.
- In case of severe steth depeg, simplest and soundest course would be to relaunch a fresh staking pool. Socializing income is to be avoided as it would effectively amount to new users supercharging the yield of past users (which would be able to withdraw their tokens once the peg is restored).

2. Informational - Consider increasing the surplus delay

Technical Details

The impact vault currently has a surplus delay of 24 hours, this in case the `steth` oracle reports an incorrect increase or decrease of the rebase ratio. In that case the next oracle update will happen 24 hours later and should fix the ratio.

But this is probably not long enough, the Lido documentation describes that the oracle can be late on the update or even miss an epoch (24 hours). In the case of a wrong ratio set by the Oracle, it is likely that the next update may be delayed or missed as Lido will need time to fix the potential problem.

In the unlikely event that the oracle report the wrong rebase ratio, it would be safer to use at least 2 or 3 days for the surplus delay so multiple epochs can happen.

Impact

Informational.

Recommendation

Consider increasing the surplus delay to multiple days so multiple epochs can happen and give a chance to the oracle to fix the rebase ratio.

Developer Response

Fixed in a4a08127c89a3f0fe98dfdeeb81a2123c916ac2c.

Conclusion

The Stake2Care protocol introduces a compelling public good application to yield generation in DeFi. Its simple yet elegant architecture reduces complexity and enhances security. The contract is gas-optimized and easy to integrate, it is currently configured for `steth` but easily adaptable to other yield-generating tokens that use a `rebase` feature.
While there are no significant issues at present, the auditor has raised concerns regarding potential `wei` loss that remains unaddressed.