PUBLIC

Code Assessment

# of the yETH Periphery Smart Contracts

August 29, 2023

Produced for

yearn

by

CHAINSECURITY

Contents

1 Executive Summary 3

2 Assessment Overview 5

3 Limitations and use of report 7

4 Terminology 8

5 Findings 9

6 Informational 10

7 Notes 11

1 Executive Summary

Dear Yearn team,

Thank you for trusting us to help Yearn with this security audit. Our executive summary provides an overview of subjects covered in our audit of the latest reviewed contracts of yETH Periphery according to Scope to support you in forming an opinion on their security risks.

Yearn implements rate providers to query the price for staked Ethereum from various staking projects.

We did not uncover any critical issues in the assessment.

In summary, we find that the codebase provides a high level of security. It is important to note that security audits are time-boxed and cannot uncover all vulnerabilities. They complement but don't replace other vital measures to secure a project.

The following sections will give an overview of the system, our methodology, the issues uncovered and how they have been addressed. We are happy to receive questions and feedback to improve our service.

Sincerely yours,

ChainSecurity

1.1 Overview of the Findings

Below we provide a brief numerical overview of the findings and how they have been addressed.

Critical-Severity Findings	0
High-Severity Findings	0
Medium-Severity Findings	0
Low-Severity Findings	0

2 Assessment Overview

In this section, we briefly describe the overall structure and scope of the engagement, including the code commit which is referenced throughout this report.

2.1 Scope

The assessment was performed on the source code files inside the yETH Periphery repository based on the documentation files. The table below indicates the code versions relevant to this report and when they were received.

V	Date	Commit Hash	Note
1	16 July 2023	12c00cf09f7703499728d71c62e7da35ed81098c	Initial Version

For the Vyper smart contracts, the compiler version `0.3.7` was chosen.

In scope were the contracts in the `contracts/providers` directory.

2.1.1 Excluded from scope

All other files and contracts which are in the repository are excluded from the scope of this assessment.

2.2 System Overview

This system overview describes the initially received version (**Version 1**) of the contracts as defined in the Assessment Overview.

Furthermore, in the findings section, we have added a version icon to each of the findings to increase the readability of the report.

Yearn implements five different contracts. All are meant to query the current price of staked Ethereum in Ethereum from various projects. No privileged roles exist and all contracts share the common interface `rate(address) -> uint256` to query and return the rate from the respective project. The projects are `Frax`, `Lido`, `Stader`, `StaFi`, `Swell` and `Tranchess`.

- The Frax rate provider contracts use `convertToAssets(UNIT)` from `0xac3E018457B222d93114458476f3E3416Abbe38F` with `UNIT` being `1e18`.
- Lido's rate provider contract uses `getPooledEthByShares(UNIT)` from address `0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84` with `UNIT` being `1e18`.
- Stader's rate provider contract uses `getExchangeRate` from address `0xF64bAe65f6f2a5277571143A24FaaFDFC0C2a737` and in the next steps calculates the rate from the returned tuple.
- StaFi's rate provider calculates the ratio of `getTotalETHBalance` and `getTotalRETHSupply` from address `0xda9726Fd1B125a3923f9D9521e28fE888091698d` to return the rate.
- Swell's rate provider uses `swETHToETHRate` directly to receive the rate from `0xf951E335afb289353dc249e82926178EaC7DEd78`.
- Tranchess's rate provider calculates the rate by dividing the return value from `getTotalUnderlying` and `getEquivalentTotalQ` from address `0x69c53679EC1C06f3275b64C428e8Cd069a2d3966`.

``

3 Limitations and use of report

Security assessments cannot uncover all existing vulnerabilities; even an assessment in which no vulnerabilities are found is not a guarantee of a secure system. However, code assessments enable the discovery of vulnerabilities that were overlooked during development and areas where additional security measures are necessary. In most cases, applications are either fully protected against a certain type of attack, or they are completely unprotected against it. Some of the issues may affect the entire application, while some lack protection only in certain areas. This is why we carry out a source code assessment aimed at determining all locations that need to be fixed. Within the customer-determined time frame, ChainSecurity has performed an assessment in order to discover as many vulnerabilities as possible.

The focus of our assessment was limited to the code parts defined in the engagement letter. We assessed whether the project follows the provided specifications. These assessments are based on the provided threat model and trust assumptions. We draw attention to the fact that due to inherent limitations in any software development process and software product, an inherent risk exists that even major failures or malfunctions can remain undetected. Further uncertainties exist in any software product or application used during the development, which itself cannot be free from any error or failures. These preconditions can have an impact on the system's code and/or functions and/or operation. We did not assess the underlying third-party infrastructure which adds further inherent risks as we rely on the correct execution of the included third-party technology stack itself. Report readers should also take into account that over the life cycle of any software, changes to the product itself or to the environment in which it is operated can have an impact leading to operational behaviors other than those initially determined in the business specification.

4 Terminology

For the purpose of this assessment, we adopt the following terminology. To classify the severity of our findings, we determine the likelihood and impact (according to the CVSS risk rating methodology).

- *Likelihood* represents the likelihood of a finding to be triggered or exploited in practice
- *Impact* specifies the technical and business-related consequences of a finding
- *Severity* is derived based on the likelihood and the impact

We categorize the findings into four distinct categories, depending on their severity. These severities are derived from the likelihood and the impact using the following table, following a standard risk assessment procedure.

Likelihood	Impact		
	High	Medium	Low
High	Critical	High	Medium
Medium	High	Medium	Low
Low	Medium	Low	Low

As seen in the table above, findings that have both a high likelihood and a high impact are classified as critical. Intuitively, such findings are likely to be triggered and cause significant disruption. Overall, the severity correlates with the associated risk. However, every finding's risk should always be closely checked, regardless of severity.

5 Findings

In this section, we describe our findings. The findings are split into these different categories:

Below we provide a numerical overview of the identified findings, split up by their severity.

Critical-Severity Findings	0
High-Severity Findings	0
Medium-Severity Findings	0
Low-Severity Findings	0

6 Informational

We utilize this section to point out informational findings that are less severe than issues. These informational issues allow us to point out more theoretical findings. Their explanation hopefully improves the overall understanding of the project's security. Furthermore, we point out findings which are unrelated to security.

6.1 Unused Import

Informational **Version 1**

CS-YRNPR-001

In the Lido rate provider, `ERC4626` is imported and never used.

7 Notes

We leverage this section to highlight further findings that are not necessarily issues. The mentioned topics serve to clarify or support the report, but do not require an immediate modification inside the project. Instead, they should raise awareness in order to improve the overall understanding.

7.1 Implementation Might Change for Proxies

Note Version 1

Multiple rate providers are proxy contracts. Their implementation might change. In consequence, incorrect rate updates or reverts might happen. Constant monitoring and updates as well as contact with the development teams of the corresponding projects might be useful for mitigation.

7.2 Providers Might Revert Instead of Returning Values

Note Version 1

When the total amount of staked tokens is null, providers behaviours can vary, some of them will return a rate of 1 to 1 with ETH while other will revert. Such a corner case should be carefully evaluated.