PERIMETER

Origin Protocol OETHVault

Fuzzing Report

March 25, 2024

Prepared By:
Rappie | Perimeter
rappie@perimetersec.io

Table of Contents

Services Provided

Perimeter has successfully delivered a comprehensive suite of services that include:

A. **Fuzzing Suite Development:** We designed and implemented a stateful fuzzing suite using Echidna and Medusa, tailor-made for the protocol and contracts in scope.

B. **Findings Reporting:** We provided thorough documentation and reporting of all findings identified throughout the engagement.

C. **Proof-of-Concept Development:** For each finding and assertion/property counterexample identified, we developed a corresponding Proof-of-Concept (PoC) to demonstrate potential vulnerabilities and their implications.

D. **Invariant Testing Assurance:** We guarantee that each invariant implemented and considered "Passed" was tested with at least 25,000,000 instances, ensuring thorough validation and reliability.

E. **Comprehensive Final Report:** We created this final report, which includes all findings and their corresponding PoCs. It also details the invariants tested, their run status, and the number of runs, providing a comprehensive overview of the engagement's outcomes.

Files in Scope

The engagement focuses on the files listed below, acquired from commit 58c56ac3770b2803c704309b28383ccb746461e8. The nSLOC data is generated using Solidity Metrics.

File	nSLOC
contracts/interfaces/IVault.sol	34
contracts/vault/OETHVaultCore.sol	80
contracts/vault/VaultCore.sol	374
Total	**488**

Files Out of Scope

Files outside the scope were not directly considered in achieving the target. However, since many of these files are utilized by those within the scope, a significant portion was indirectly covered.

Methodology

The primary goal of this engagement was to identify potential issues in refactoring the Vault contract. This refactoring effort aims to simplify the minting and redeeming behavior of OETH tokens, thereby adding the ability to eliminate redemption fees. The current code is known to contain several acceptable rounding errors. Origin Protocol requested that we investigate whether an attacker could exploit these errors to extract value from the system, particularly in scenarios where redemption fees are at zero percent.

The main threats under investigation were:

1. **Misappropriation of Funds:** We investigated the possibility that an attacker could alter the distribution of OETH, potentially leading to the misappropriation of funds by other users. The attack could occur if a malicious actor manipulates the total OETH supply to exceed the Vault assets.

2. **Unfair Yield Distribution:** There were concerns that an attacker could obtain a disproportionate share of the yield from the Vault. Our investigation aimed to verify whether this was the case. After this was confirmed, we defined acceptable tolerance levels and created a Proof of Concept to demonstrate how this manipulation could occur.

To achieve the goals set above, we designed and implemented multiple invariants to test the correct functioning of the OETH and OETHVault contracts. These invariants allowed us to thoroughly test actions related to using the protocol. We established acceptable tolerance levels on a case-by-case basis to manage any rounding errors that may occur.

Using Echidna's optimization mode, we identified the maximum rounding errors and determined a suitable tolerance threshold. We then integrated this into the fuzzing suite through collaboration with Origin Protocol. This approach ensured precise and realistic testing, recognizing that rounding errors are unavoidable.

Invariants

We created many tests to verify the integrity of several system invariants described in the table below. Throughout the engagement, these invariants were assessed for a total of 130,000,000+ runs with a prepared Echidna & Medusa fuzzing suite.

To enhance understanding and readability, we describe the invariants as though they were tested with 100% precision. However, due to rounding errors, achieving this level of accuracy was not feasible for all invariants.

The table below lists all invariants that are part of this engagement.

Invariant	Description	Tested	Passed	# Runs
OETH-01	Transfering OETH does not unexpectedly revert	✅	✅	130M+
OETH-02	Opting in to rebase does not unexpectedly revert	✅	✅	130M+
OETH-03	Opting out of rebase does not unexpectedly revert	✅	✅	130M+
VMINT-01	Actor WETH balance decreases by amount minted after successful mint	✅	✅	130M+
VMINT-02	Actor OETH balance increases by amount minted after successful mint	✅	✅	130M+
VMINT-03	Vault WETH balance increases by amount minted after successful mint	✅	✅	130M+
VMINT-04	Minting OETH does not unexpectedly revert	✅	✅	130M+
VREDEEM-01	Actor WETH balance increases by amount redeemed after successful redeem	✅	✅	130M+
VREDEEM-02	Actor OETH balance decreases by amount redeemed after successful redeem	✅	✅	130M+
VREDEEM-03	Vault WETH balance decreases by amount redeemed after successful redeem	✅	✅	130M+
VREDEEM-04	Redeeming OETH does not unexpectedly revert	✅	✅	130M+
VREDEEM-05	Actor OETH balance is zero after successfully redeeming all	✅	✅	130M+
VREBASE-01	Donating WETH to the Vault does not unexpectedly revert	✅	✅	130M+

Invariant	Description	Tested	Passed	# Runs
VREBASE-02	Rebasing never decreases OETH balance for any actor	✅	✅	130M+
VREBASE-03	Rebasing vault does not unexpectedly revert	✅	✅	130M+
GLOBAL-01	The sum of WETH held by all actors never exceeds the sum of their WETH starting balances	✅	✅	130M+
GLOBAL-02	The sum of OETH held by all actors never exceeds the sum of their WETH starting balances	✅	✅	130M+
GLOBAL-03	The total amount of generated yield equals the total amount of WETH donated to the Vault	✅	✅	130M+
GLOBAL-04	The sum of all starting balances minus the total amount of WETH donated equals the sum of all WETH and OETH balances minus the total amount of yield generated	✅	✅	130M+
GLOBAL-05	The total supply of OETH never exceeds the total supply of WETH	✅	✅	130M+
GLOBAL-06	The Vault WETH balance never exceeds the total the amount of OETH held by all actors and outsiders	✅	✅	130M+
GLOBAL-07	Any actor can always redeem all OETH	✅	✅	130M+

Findings

Rounding errors in subsequent mint, rebase, and redemption can lead to stolen yield

Severity: High Risk

Description: Due to rounding errors in the amount minted, amount redeemed, and yield gained after rebasing, a malicious actor could receive a yield in OETH redeemable for more WETH than minted to the vault before the rebase. This scenario enables an attacker to steal a percentage of the yield from other protocol users and a small amount of WETH from the vault.

A potential real-world attack would be to perform a sandwich attack around the rebase transaction. As the first transaction, an attacker mints a significant amount of OETH just before the rebase. Immediately afterward, the attacker redeems all, including the accrued yield.

Proof of concept: The proof of concept is available here.

Recommendation: Remove rounding errors from the protocol or keep redemption fees sufficiently high.

Tolerances

Rounding errors that cause multiple invariants to break have been identified during our investigation. These issues exist in the current codebase and fall within acceptable limits.

To address this, we collaborated with Origin Protocol to define acceptable tolerances for each affected invariant. The table below outlines these invariants along with their respective tolerances.

All identified rounding errors fall within an acceptable tolerance, allowing for the safe deployment of the refactored code. However, redemption fees should be sufficiently high to safeguard against the issue: Rounding errors in subsequent mint, rebase, and redemption can lead to stolen yield.

Invariant	Description	Tolerance
`VMINT-01`	Actor WETH balance decreases by amount minted after successful mint	1
`VMINT-02`	Actor OETH balance increases by amount minted after successful mint	1
`VMINT-03`	Vault WETH balance increases by amount minted after successful mint	1
`VREDEEM-01`	Actor WETH balance increases by amount redeemed after successful redeem	1
`VREDEEM-02`	Actor OETH balance decreases by amount redeemed after successful redeem	1
`VREDEEM-03`	Vault WETH balance decreases by amount redeemed after successful redeem	1
`VREDEEM-05`	Actor OETH balance is zero after successfully redeeming all	1
`VREBASE-02`	Rebasing never decreases OETH balance for any actor	1
`GLOBAL-01`	The sum of WETH held by all actors never exceeds the sum of their WETH starting balances	10,000
`GLOBAL-03`	The total amount of generated yield equals the total amount of WETH donated to the Vault	20,000,000
`GLOBAL-04`	The sum of all starting balances minus the total amount of WETH donated equals the sum of all WETH and OETH balances minus the total amount of yield generated	10

Invariant	Description	Tolerance
`GLOBAL-06`	The Vault WETH balance never exceeds the total the amount of OETH held by all actors and outsiders	20,000,000