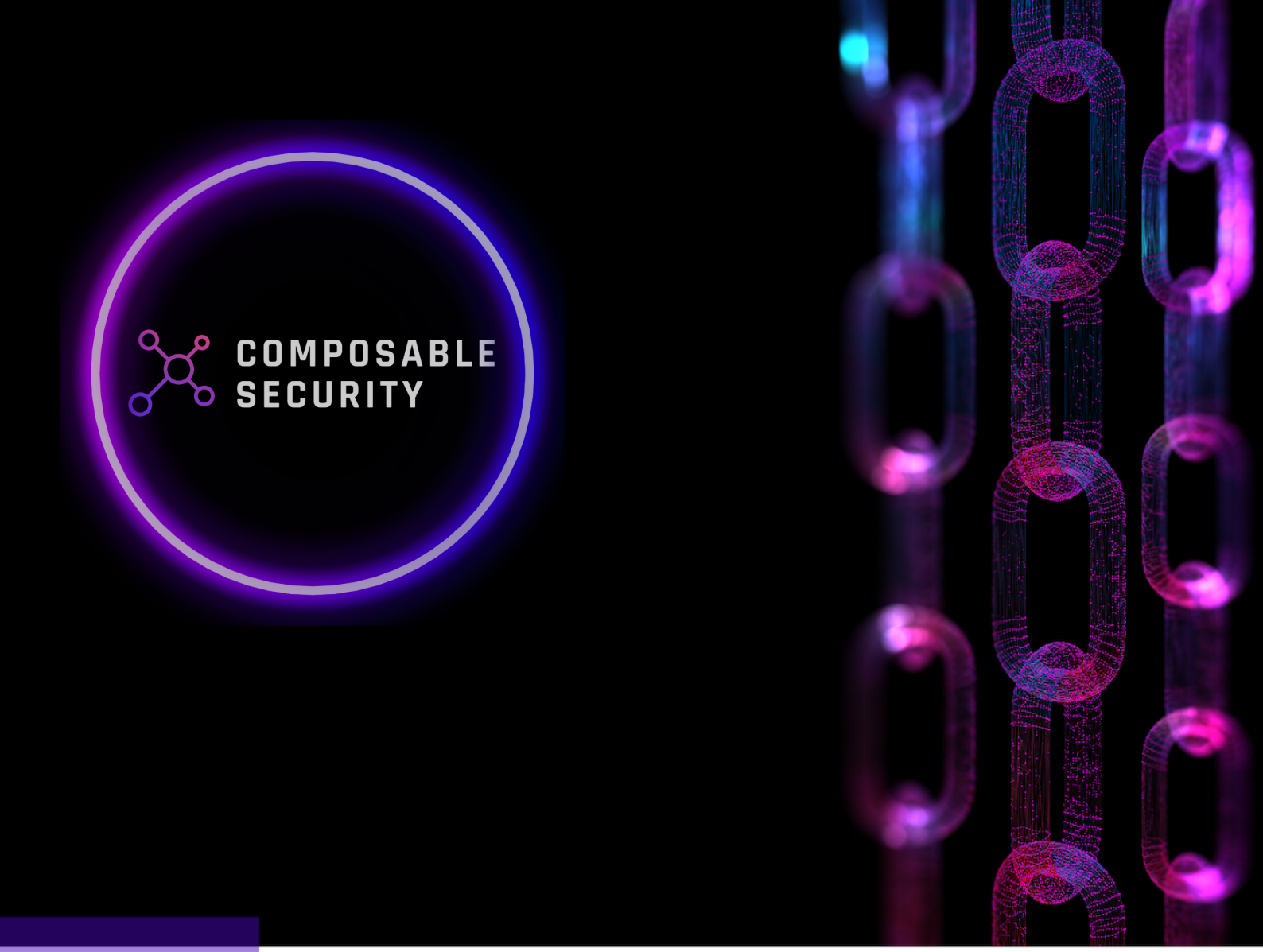

COMPOSABLE SECURITY

REPORT

Security consultation for Lido

Prepared by: Composable Security
Report ID: LDO-10705137
Test time period: 2025-09-27 - 2025-09-27
Report date: 2025-10-01
Version: 1.0
Visit: composable-security.com

Contents

1. Security consultation summary (2025-09-27)

1.1. Subject of consultation

The **Composable Security** team was commissioned by the **Lido** to conduct a security review of fix for an issue that did not allow to build new report by Accounting Oracle.

This fix resolves a security vulnerability that caused an error and prevented the report from being created when there were more than 1 iteration of the `calculate_finalization_batches` function when the oracle was fetching finalization batches.

1.2. Scope

The subjects of the test were changes since the retest of previous audit of **Oracle V6 upgrade** → 6.0.2.

GitHub repository:
https://github.com/lidofinance/lido-oracle
CommitID: 10705137a8be0c4b9f4fb52bfdb8ece12d77cb69

1.3. Root cause

The `calculate_finalization_batches` function is cached using the `lru_cache` decorator. The limitation of caching is that the function parameters cannot be lists; therefore, each list must be changed to tuple. This is also why the `BatchState` dataclass, which is the result of the `calculate_finalization_batches` function, has the `batches` type is defined as a tuple.

```
102 @dataclass
103 class BatchState:
104     remaining_eth_budget: int
105     finished: bool
106     batches: tuple[int, ...] # <-----
107     batches_length: int
```

The `calculate_finalization_batches` function calls the on-chain `calculateFinalizationBatches` function and returns the updated state which is mapped to the `BatchState` dataclass.

However, it does not return a `tuple` type for the `batches` field, but a `list`. Therefore, the list is assigned to the field `batches`, and the function is called again with the updated state (in the parameter `batch_state`). This leads to calling the function with a list parameter, and caching raises an error.

1.4. The fix

The proposed solution removes the `lru_cache` decorator and disables caching for the `calculate_finalization_batches` function. Additionally, the type of field `batches` is changed to `list` to correctly reflect the returned type.

1.5. Consultation results

The fix effectively addresses the vulnerability because the caching function is no longer called when the `calculate_finalization_batches` function is executed.

We also verified that no other cached function accepts parameters of type `list`.

1.6. Deployments

After the security review conducted, we verified that the Dockerfile used to build an image uses the reviewed source code. The published image with the manifest digest **sha256:1b0501724c9c3e00dc6c03b663dfccc5af545a8bacc9543bbc5456d408d6f098** corresponds to the commitID: **10705137a8be0c4b9f4fb52bfdb8ece12d77cb69**.

1.7. Disclaimer

Security consultation **IS NOT A SECURITY WARRANTY**.

During the review, the Composable Security team makes every effort to detect any occurring problems and help to address them. However, it is not allowed to treat the report as a security certificate and assume that the project does not contain any vulnerabilities. Securing applications is a multi-stage process, starting from threat modeling, through development based on best practices, security reviews and formal verification, ending with constant monitoring and incident response.

Therefore, we encourage the implementation of security mechanisms at all stages of development and maintenance.

Damian Rusinek

Smart Contracts Auditor

@drdr_zz

damian.rusinek@composable-security.com

Paweł Kuryłowicz

Smart Contracts Auditor

@wh01s7

pawel.kurylowicz@composable-security.com