Smart Contract Security Audit Report

Gearbox

1. Contents

1. Contents 2
2. General Information 3
2.1. Introduction 3
2.2. Scope of Work 3
2.3. Threat Model 4
2.4. Weakness Scoring 4
2.5. Disclaimer 4
3. Summary 5
3.1. Suggestions 5
4. General Recommendations 6
4.1. Security Process Improvement 6
5. Findings 7
5.1. Potential DOS of WETH swaps 7
5.2. Incorrect comment 7
5.3. Gas optimization 8
6. Appendix 10
6.1. About us 10

2. General Information

This report contains information about the results of the security audit of the Gearbox (hereafter referred to as "Customer") smart contracts, conducted by Decurity in the period from 29/01/2025 to 31/01/2025.

2.1. Introduction

Tasks solved during the work are:

- Review the protocol design and the usage of 3rd party dependencies,
- Audit the contracts implementation,
- Develop the recommendations and suggestions to improve the security of the contracts.

2.2. Scope of Work

The audit scope included the contracts in the following repositories: integrations-v3 (commits e055acf and 6bd120). Re-testing was done for the commits 43c85fa and 57e535e. After the re-testing Gearbox has introduced minor naming changes in the commit 9e56cb.

The following contracts have been tested:

- integrations-v3-balancer-v3-router/contracts/adapters/balancer/BalancerV3RouterAdapter.sol
- integrations-v3-balancer-v3-router-3.1/contracts/adapters/balancer/BalancerV3RouterAdapter.sol
- balancer-v3-gateway-3.0/contracts/helpers/balancer/BalancerV3RouterGateway.sol
- balancer-v3-gateway-3.1/contracts/helpers/balancer/BalancerV3RouterGateway.sol

2.3. Threat Model

The assessment presumes actions of an intruder who might have capabilities of any role (an external user, token owner, token service owner, a contract). The centralization risks have not been considered upon the request of the Customer.

The main possible threat actors are:

- User
- Balancer protocol
- Protocol Owner

2.4. Weakness Scoring

An expert evaluation scores the findings in this report, an impact of each vulnerability is calculated based on its ease of exploitation (based on the industry practice and our experience) and severity (for the considered threats).

2.5. Disclaimer

Due to the intrinsic nature of the software and vulnerabilities and the changing threat landscape, it cannot be generally guaranteed that a certain security property of a program holds.

Therefore, this report is provided “as is” and is not a guarantee that the analyzed system does not contain any other security weaknesses or vulnerabilities. Furthermore, this report is not an endorsement of the Customer’s project, nor is it an investment advice.

That being said, Decurity exercises best effort to perform their contractual obligations and follow the industry methodologies to discover as many weaknesses as possible and maximize the audit coverage using the limited resources.

3. Summary

During audit we have detected multiple informational issues.

3.1. Suggestions

The table below contains the discovered issues, their risk level, and their status as of February 17, 2025.

Table. Discovered weaknesses

Issue	Contract	Risk Level	Status
Potential DOS of WETH swaps	balancer-v3-gateway-3.1/contracts/helpers/balancer/BalancerV3RouterGateway.sol, balancer-v3-gateway-3.0/contracts/helpers/balancer/BalancerV3RouterGateway.sol	Info	Fixed
Incorrect comment	integrations-v3-balancer-v3-router-3.1/contracts/adapters/balancer/BalancerV3RouterAdapter.sol, integrations-v3-balancer-v3-router/contracts/adapters/balancer/BalancerV3RouterAdapter.sol	Info	Fixed
Gas optimization	integrations-v3-balancer-v3-router/contracts/adapters/balancer/BalancerV3RouterAdapter.sol	Info	Fixed

4. General Recommendations

This section contains general recommendations on how to improve overall security level.

The Findings section contains technical recommendations for each discovered issue.

4.1. Security Process Improvement

The following is a brief long-term action plan to mitigate further weaknesses and bring the product security to a higher level:

- Keep the whitepaper and documentation updated to make it consistent with the implementation and the intended use cases of the system,
- Perform regular audits for all the new contracts and updates,
- Ensure the secure off-chain storage and processing of the credentials (e.g. the privileged private keys),
- Launch a public bug bounty campaign for the contracts.

5. Findings

5.1. Potential DOS of WETH swaps

Risk Level: Info

Status: Fixed in pulls 156 and 155.

Contracts:

- balancer-v3-gateway-3.1/contracts/helpers/balancer/BalancerV3RouterGateway.sol,
- balancer-v3-gateway-3.0/contracts/helpers/balancer/BalancerV3RouterGateway.sol

Description:

The Gateway contract lacks a `receive` function, which causes transactions to revert when the Balancer router attempts to transfer ETH during swaps where WETH is the `tokenIn`. This vulnerability enables a DOS attack by allowing an attacker to donate small amounts of ETH to the Balancer router before WETH swaps are executed.

Remediation:

Consider adding receive function to the Gateway contract. Admin controlled function to withdraw ETH can also be added.

5.2. Incorrect comment

Risk Level: Info

Status: Fixed in the commit b83143e.

Contracts:

- integrations-v3-balancer-v3-router-3.1/contracts/adapters/balancer/BalancerV3RouterAdapter.sol,
- integrations-v3-balancer-v3-router/contracts/adapters/balancer/BalancerV3RouterAdapter.sol

Description:

The comment for `EnumerableSet.AddressSet internal _allowedPools;` states: `Set of all pools that were ever allowed`. This comment is incorrect since the `_allowedPools` list is mutable and pools can be removed from it.

Remediation:

Consider updating the comment to accurately reflect the variable's purpose.

5.3. Gas optimization

Risk Level: Info

Status: Fixed in the commit b83143e.

Contracts:

- integrations-v3-balancer-v3-router/contracts/adapters/balancer/BalancerV3RouterAdapter.sol,

Description:

Array reads by index should be avoided when you use the variable more than 1 time. It is cheaper to make a copy and use that inside the function.

```
function setPoolStatusBatch(address[] calldata pools, bool[] calldata
statuses)
        external
        override
        configuratorOnly
    {
        uint256 len = pools.length;
        if (len != statuses.length) revert InvalidLengthException();
        unchecked {
            for (uint256 i; i < len; ++i) {
                // @gas create a copy of pools[i] and statuses[i] to avoid
multiple reading
                _poolStatus[pools[i]] = statuses[i];
                if (statuses[i]) {
                    _allowedPools.add(pools[i]);
                } else {
                    _allowedPools.remove(pools[i]);
                }
                emit SetPoolStatus(pools[i], statuses[i]);
            }
        }
    }
```

Remediation:

Consider caching `pools[i]` and `statuses[i]` variables in the same way as it is done in the `balancer-v3.1`

6. Appendix

6.1. About us

The Decurity team consists of experienced hackers who have been doing application security assessments and penetration testing for over a decade.

During the recent years, we've gained expertise in the blockchain field and have conducted numerous audits for both centralized and decentralized projects: exchanges, protocols, and blockchain nodes.

Our efforts have helped to protect hundreds of millions of dollars and make web3 a safer place.