Treehouse tETH / Gearbox kpkwsteth Audit Report

Nov 04, 2025

Table of Contents

Summary

This report has been prepared for Treehouse smart contract, to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Overview

Project Summary

Project Name	**Treehouse**
Codebase	**https://github.com/treehouse-gaia/tETH-protocol**
Commit	**287554e4277cf53b84237823ed9cefe5e11d2036**
Language	**Solidity**

Audit Summary

Delivery Date	**Nov 04, 2025**
Audit Methodology	**Static Analysis, Manual Review**
Total Isssues	**1**

[WP-M1] `StrategyExecutor.executors` allows arbitrary pool specification through calldata parameters, enabling malicious pools to drain strategy assets

Medium

Issue Description

The pool parameter in `GearboxDepositV31` and `GearboxRedeemV31` is entirely determined by `_actionCalldata` in `StrategyExecutor.executeOnStrategy(uint _strategyId, bytes4[] calldata _actionIds, bytes[] calldata _actionCalldata, uint8[][] memory _paramMapping)` . Unlike other Actions, they lack constant or immutable variable restrictions and directly use the pool parameter passed in.

If the pool is not the originally intended pool for `GearboxDepositV31` and `GearboxRedeemV31` , it can be exploited to steal share tokens or any asset tokens (the token in `GearboxDepositV31` L45 is determined by `pool.asset()`).

https://github.com/treehouse-gaia/tETH-protocol/blob/0625597754a24a6b805253a279f38d2bd62a5e09/contracts/strategy/actions/gearbox/GearboxDepositV31.sol

```
@@ 1,6 @@
7
8  /// @title Deposit token into gearbox pool
9  contract GearboxDepositV31 is ActionBase {
@@ 10,24 @@
25
26   /// @inheritdoc ActionBase
27   function executeAction(
28     bytes calldata _callData,
29     uint8[] memory _paramMapping,
30     bytes32[] memory _returnValues
31   ) public payable virtual override returns (bytes32) {
32     Params memory params = parseInputs(_callData);
33     params.assetAmount = _parseParamUint(params.assetAmount, _paramMapping[0],
_returnValues);
34     (uint shares, bytes memory logData) = _deposit(params.pool,
params.assetAmount);
```

```
35      emit ActionEvent(NAME, logData);
36      return bytes32(shares);
37    }
38
39    ////////////////////////// ACTION LOGIC //////////////////////////
40
41    /// @notice User deposits tokens into gearbox pool
42    /// @param pool The address of the deposit vault
43    /// @param assetAmount Amount of tokens to be deposited
44    function _deposit(address pool, uint assetAmount) internal returns (uint, bytes
      memory) {
45      IPoolV3(pool).asset().approveToken(pool, assetAmount);
46
47      uint shares = IPoolV3(pool).deposit(assetAmount, address(this));
48      bytes memory logData = abi.encode(pool, assetAmount, shares);
49      return (shares, logData);
50    }
51
52    function parseInputs(bytes memory _callData) public pure returns (Params memory
      params) {
53      params = abi.decode(_callData, (Params));
54    }
55  }
```

https://github.com/treehouse-gaia/tETH-protocol/blob/0625597754a24a6b805253a279f38d2bd62a5e09/contracts/strategy/actions/gearbox/GearboxRedeemV31.sol

```
1   // SPDX-License-Identifier: MIT
    @@ 2,6 @@
7
8   /// @title Redeem gearbox receipt token
9   contract GearboxRedeemV31 is ActionBase {
    @@ 10,24 @@
25
26    /// @inheritdoc ActionBase
27    function executeAction(
28      bytes calldata _callData,
29      uint8[] memory _paramMapping,
```

```
30      bytes32[] memory _returnValues
31    ) public payable virtual override returns (bytes32) {
32      Params memory params = parseInputs(_callData);
33      params.shareAmount = _parseParamUint(params.shareAmount, _paramMapping[0],
    _returnValues);
34      (uint assets, bytes memory logData) = _redeem(params.pool,
    params.shareAmount);
35      emit ActionEvent(NAME, logData);
36      return bytes32(assets);
37    }
38
39    ////////////////////////// ACTION LOGIC //////////////////////////
40
41    /// @notice User redeems vault share for asset tokens
42    /// @param pool The address of the deposit vault
43    /// @param shareAmount Amount of shares to redeem for assets
44    function _redeem(address pool, uint shareAmount) internal returns (uint, bytes
    memory) {
45      uint assets = IPoolV3(pool).redeem(shareAmount, address(this), address(this));
46      bytes memory logData = abi.encode(pool, shareAmount, assets);
47      return (assets, logData);
48    }
49
50    function parseInputs(bytes memory _callData) public pure returns (Params memory
    params) {
51      params = abi.decode(_callData, (Params));
52    }
53  }
```

Recommendation

Consider adding a pool whitelist maintained by Treehouse, managed by a role with higher privileges than StrategyExecutor (e.g., multisig).

Status

✓ Fixed

Appendix

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by WatchPug; however, WatchPug does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Smart Contract technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.