Nereus Staking Contracts Code Audit by Ambisafe Inc.

June, 2022

Artem Martiukhin, Oleksii Matiiasevych

1. **Introduction.** Nereus Finance. requested Ambisafe to perform a code audit of the contracts implementing Nereus Protocol. The contracts in question can be identified by the following git commit hash:

   `73d1a65c3a7fbf0f12834b30855e0fbe5857f30b`

   The scope of the audit is NXUSD, LPAToken, PermissionManager and CauldronV2 contracts.

   During the initial code audit, Nereus Finance team applied a number of updates which can be identified by the following git commit hash:

   `0a6694fb41b2d03949b6fbe1968791c3442281ac`

   Additional verification was performed after that.

2. **Disclaimer.** The code audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts for any specific purpose, or their bugfree status. The code audit documentation below is for internal management discussion purposes only and should not be used or relied upon by external parties without the express written consent of Ambisafe.

3. **Executive Summary.** All the initially identified, minor and above, severity issues were **fixed** and are not present in the **final version** of the contracts. There are **no** known compiler bugs for the specified compiler version (0.6.12), that might affect the contracts' logic. There were 1 critical, 0 major, 0 minor, 0 informational and optimizational issues identified in the initial version of the contracts. Issues found in the contract were not present in the final version. They are described below for historical purposes.
   Modifications of the CauldronV2 mainly consist of introducing a liquidator role, making

the **liquidate()** function only accessible to certain addresses, and collecting additional liquidation fees for sSpell holders.

4. **Critical Bugs and Vulnerabilities.** Single critical vulnerability was identified that allowed anyone to reinitialize the LPAToken contract.

5. **Initial Line By Line Review.**

5.1. LPAToken, line 36. **Critical**, the **initialize()** function could be executed by anyone, after the setup is done.

Oleksii Matiiasevych