VENUS

Venus Security Analysis

by Pessimistic

This report is public

February 29, 2024

Abstract

In this report, we consider the security of smart contracts of Venus project. Our task is to find and describe security issues in the smart contracts of the platform.

Disclaimer

Although a security audit is one of the most crucial steps to overall project security, it does not give any guarantees on the security of the code. A single audit cannot be considered enough. We strongly encourage our customers to take additional measures to achieve maximum project security. One can check out our public documentation on how to protect a project beyond the code audit. Besides, a security audit is not investment advice.

On code diff security analysis

Our general policy is that we do not audit code diffs and do the full audits of the codebase instead.

For this project, we did not have a possibility to audit the full codebase, only two pull requests. We can not provide our usual quality of work within these limitations, but the security review of this limited scope still increases the confidence level. This approach is not enough on its own, but it may be used as a supplement for the rest of the security measures. It must be explicitly stated that during this work, we assume that the initial codebase is completely secure and the update does not break any system invariants.

Generally, auditing only the updated parts of the code without the team being familiar with the rest of the codebase is often counterproductive and creates a dangerous illusion of proper risk management. Many aspects and security invariants of the system fall out of the scope of such security review. To provide the work of our usual quality standard and the same level of confidence, we need to audit the whole project, which means the full review of the unchanged parts of the code, too, unless we have audited that code in the last six months.

However, in some cases, there is a time or budget constraint on the side of our customer, which makes the full audit impossible. In such cases, we provide a best-effort security analysis within this limited time to increase the confidence level of some limited scope (usually some code update). This analysis can not provide the same quality level as our standard procedure.

Summary

In this report, we considered the security of Venus smart contracts. We described the audit process in the section below.

The initial audit showed only several issues of low severity. They do not endanger project security. All the tests passed. The documentation is not public.

After the initial audit, the codebase was updated. The developers fixed all issues of low-severity. They increased the number of tests and the code coverage.

General recommendations

We recommend making the documentation publicly accessible.

Project overview

Project description

For the audit, we were provided with the following pull requests:

- Isolated-pools, commit: bb127763e9a97d1e50427a7a247888c1b2cbfeae

  The scope of the PR included:

  - **contracts/Gateway/NativeTokenGateway.sol**;
  - **contracts/Comptroller.sol**;
  - **contracts/ComptrollerStorage.sol**;
  - **contracts/VToken.sol**;
  - **contracts/VTokenInterfaces.sol**;
  - **ComptrollerInterface.sol**;
  - **ErrorReporter.sol**.

  All 434 tests pass successfully. The code coverage is 96.22%.

  The PR includes: 315 insertions(+), 10 deletions(-);

- Venus-Protocol, commit becfe891329b8c93f46e968051721848d6d05253

  The scope of the PR included:

  - **contracts/Tokens/VTokens/VBep20.sol**;
  - **contracts/Tokens/VTokens/VToken.sol**.

  All 629 tests pass successfully. The code coverage is 65.66%.

  The PR includes: 66 insertions(+), 15 deletions(-).

The documentation for the project included:

- **Audit scope-NativeTokenGateway (2024 02 06).pdf** with the following checksum: `5b045c3808598ca28c857e5a4c15b870ef73bbd76c0897bfa241f0b97c27e951`.

Codebase update #1

After the initial audit, the codebase was updated, and we were provided with commits for two pull requests:

- Isolated-pools: 4211d2dd29d2ca2e64b5349b40a222223e4590ba. All 434 tests passed. The code coverage was 96.22%.
- Venus-Protocol: dbd4edcd43bee1a80b57fe034259653eef158e92. All 634 tests passed. The code coverage was 65.67%

The developers fixed all low-severity issues.

Audit process

We started the audit on February 20, 2024 and finished on February 22, 2024. It was not a full audit, just the pull requests of the scope from the Project Description. Since we did not audit the whole project, there was a big chance to miss bugs. That is why we wrote the Disclaimer.

We inspected the materials provided for the audit. Then, we contacted the developers for an introduction to the project. After a discussion, we performed preliminary research and started the review.

We manually analyzed all the contracts within the scope of the audit and checked their logic. Among other, we verified the following properties of the contracts:

- Only the approved delegatee is able to invoke operations on behalf of the user;
- The dust upon repayment is handled correctly, thus there is no possibility of leftovers;
- There is no possibility of corruption during deployment on L2s;
- The codebase does not contain unfinished functions or contracts;
- Storage management is accurately maintained throughout the upgrade process;
- Functions revert with informative messages when incorrect arguments are supplied;
- A delegatee cannot manipulate a position and then liquidate it.

We scanned the project with the following tools:

- Static analyzer Slither;
- Our plugin Slitherin with an extended set of rules;
- Semgrep rules for smart contracts. We also sent the results to the developers in the text file.

We ran tests and calculated the code coverage.

We combined in a private report all the verified issues we found during the manual audit or discovered by automated tools.

We made the recheck on February 27-29, 2024. We checked whether the previous issues were fixed and the code updates. Also, we ran tests, calculated the code coverage and updated the report.

Manual analysis

The contracts were completely manually analyzed, their logic was checked. Besides, the results of the automated analysis were manually verified. All the confirmed issues are described below.

Critical issues

Critical issues seriously endanger project security. They can lead to loss of funds or other catastrophic consequences. The contracts should not be deployed before these issues are fixed.

The audit showed no critical issues.

Medium severity issues

Medium severity issues can influence project operation in the current implementation. Bugs, loss of potential income, and other non-critical failures fall into this category, as well as potential problems related to incorrect system management. We highly recommend addressing them.

The audit showed no issues of medium severity.

Low severity issues

Low severity issues do not directly affect project operation. However, they might lead to various problems in future versions of the code. We recommend fixing them or explaining why the team has chosen a particular option.

L01. Typo in the comment (fixed)

The `reedemer` word of the `We invoke _doTransferOut for the redeemer and the redeemAmount` comment should be replaced with the `receiver` in the `vToken._redeemFresh` function. This issue occurs at line 988 of the isolated-pools and at line 980 of the venus-protocol repositories.

The issues have been fixed and are not present in the latest version of the code.

L02. Caching the state variable (fixed)

The following functions `sweepNative()` and `sweepToken()` make redundant warm access call to an `owner()`, which could be cached to save gas.

The issues have been fixed and are not present in the latest version of the code.

Notes

N01. Front-running the delegatee update (commented)

During the `Comptroller.updateDelegate` call, a user can revoke delegatee rights. This transaction might be front-runned by a "spender". Consequently, the "spender" can redeem tokens up to the limit permitted by the `Comptroller._checkRedeemAllowed` function in the `Comptroller.preRedeemHook` call. While this action does not result in position liquidation, it can bring it closer to that point.

> *Comment from the developers: Acknowledged.*

This analysis was performed by Pessimistic:

Daria Korepanova, Senior Security Engineer
Yhtyyar Sahatov, Security Engineer
Rasul Yunusov, Security Engineer
Konstantin Zherebtsov, Business Development Lead
Irina Vikhareva, Project Manager
Alexander Seleznev, Founder

February 29, 2024