OpenZeppelin | security

Venus Liquidator Audit

July 20, 2023

This security assessment was prepared by OpenZeppelin.

Table of Contents

Summary

Type	DeFi
Timeline	From 2023-06-28 To 2023-07-05
Languages	Solidity

Total Issues	3 (3 resolved)
Critical Severity Issues	0 (0 resolved)
High Severity Issues	0 (0 resolved)
Medium Severity Issues	0 (0 resolved)
Low Severity Issues	0 (0 resolved)
Notes & Additional Information	3 (3 resolved)

Scope

We audited the VenusProtocol/venus-protocol repository at the 8016d5fe5604f62ab67c6f87234e2f92279785d3 commit. More specifically, we audited pull request #241 and pull request #299.

In scope were the following contracts:

```
contracts
└── Liquidator
    ├── Liquidator.sol
    └── LiquidatorStorage.sol
```

System Overview

The Venus Protocol combines mintable stablecoins, like MakerDAO's Dai, with algorithmic lending markets, like Compound, into a single platform built on the BNB chain. Venus's stablecoin is called VAI and is native to their platform. Any BEP-20 token can also be used once it has a listed market. When listed, a BEP-20 will have an equivalent vToken which will be used to execute any actions, like borrowing, in the protocol. Similarly, BNB and wBNB can be used in the protocol once deposited into vBNB.

This audit encompassed updates to the Liquidation contract of the Venus protocol. The liquidation contract allows users the ability to liquidate positions that have gone below their collateralization ratio. It also allows for some administration around which individuals can liquidate as well as payout ratios from liquidation rewards.

Both pull request #241 and pull request #299 were reviewed in this audit. Some of the additions include the introduction of the `AccessControlManager` for access control rather than the `OnlyOwner` modifier, VAI liquidation requirements, and the redemption of vTokens sent to `protocolShareReserve`.

Security Model and Trust Assumptions

Privileged Roles

Access control has shifted from `OnlyOwner` to the `AccessControlManager`. This allows the Venus team to use governance contracts to administrate how each function can be called. Some functions, now controlled by the governance, have the ability to negatively impact the protocol if used in a malicious way. Since malicious behavior from the governance is against its own interests, this is not considered an issue.

Notes & Additional Information

N-01 Incorrect Docstring

The docstring for `resumeForceVAILiquidate` is incorrect. Consider replacing "pause" with "resume".

Update: *Resolved in pull request #305 at commit e4832bd.*

N-02 Improper Use of Named Return Variables

Named return variables are a way to declare variables that are meant to be used within a function's body for the purpose of being returned as the function's output. They are an alternative to explicit in-line `return` statements.

Specifically, in `Liquidator.sol`, both `_distributeLiquidationIncentive` and `_splitLiquidationIncentive` have named return variables that are explicitly returned. Additionally, `_transferBep20` returns something other than its named return variable.

Consider either using or removing any unused named return variables, as well as omitting returns for already named return variables.

Update: *Resolved in pull request #305 at commit a1ebea6.*

N-03 Using `int`/`uint` Instead of `int256`/`uint256`

Within `Liquidator.sol` there are multiple instances where `int`/`uint` are used instead of `int256`/`uint256`. For instance:

- On line 347 in `Liquidator.sol`
- On line 444 in `Liquidator.sol`

In favor of explicitness, consider replacing all instances of `int`/`uint` with `int256`/`uint256`.

Update: *Resolved in pull request #305 at commit 460db5a.*

Conclusions

The Liquidator update improves access control and liquidation flow. Three notes were included to improve the overall quality of the codebase. Some changes were proposed regarding smart contract security best practices.

Appendix

Monitoring Recommendations

While audits help in identifying code-level issues in the current implementation and potentially the code deployed in production, the Venus team is encouraged to consider incorporating monitoring activities into their operations. Ongoing monitoring of deployed contracts helps identify potential threats and issues affecting production environments.

To ensure no unexpected administrative actions are occurring, and to validate that correct values were used, consider monitoring all permissioned administrative events. In particular, consider monitoring:

- `restrictLiquidation`
- `unrestrictLiquidation`
- `addToAllowlist`
- `removeFromAllowlist`
- `setTreasuryPercent`
- `setProtocolShareReserve`

In addition, consider monitoring liquidation events to ensure positions are under-collateralized.