OpenZeppelin | security

Venus SwapRouter Audit

June 16, 2023

This security assessment was prepared by OpenZeppelin.

Table of Contents

Table of Contents 2

Summary 3

Scope 4

System Overview 5

Security Model and Trust Assumptions 5

Privileged Roles 5

Low Severity 6

L-01 Missing Docstrings 6

L-02 Locked BNB in Contract 6

Notes & Additional Information 7

N-01 Misleading Docstrings 7

N-02 Naming Can Be Improved 7

N-03 Some Convenience Functions Are Missing 7

N-04 Confusing Use of ETH and BNB in Comments and Function Names 8

N-05 Lack of SPDX License Identifiers 8

N-06 PancakeRouter Functions' Code Can Be Reused 9

Conclusions 10

Appendix 11

Monitoring Recommendations 11

Summary

Type	DeFi
Timeline	From 2023-05-15 To 2023-06-01
Languages	Solidity

Total Issues	8 (7 resolved)
Critical Severity Issues	0 (0 resolved)
High Severity Issues	0 (0 resolved)
Medium Severity Issues	0 (0 resolved)
Low Severity Issues	2 (2 resolved)
Notes & Additional Information	6 (5 resolved)

Scope

We audited the VenusProtocol/venus-protocol repository at the e5f112ac6f8cb9e5dcf760ad3626c9f3fc43609c commit.

In scope were the following contracts:

```
contracts
└── Swap
    ├── IRouterHelper.sol
    ├── RouterHelper.sol
    ├── SwapRouter.sol
    ├── interfaces
    │   ├── CustomErrors.sol
    │   ├── IPancakePair.sol
    │   ├── IPancakeSwapV2Factory.sol
    │   ├── IPancakeSwapV2Router.sol
    │   ├── IVBNB.sol
    │   ├── IVtoken.sol
    │   ├── IWBNB.sol
    │   └── InterfaceComptroller.sol
    └── lib
        ├── PancakeLibrary.sol
        └── TransferHelper.sol
```

System Overview

Venus SwapRouter is a tailored adaptation of the PancakeSwap V2 PancakeRouter which broadens the range of user interactions by adding Venus markets. Its usefulness comes from the ability to carry out operations like supplying collateral and repaying debt starting with a token that may differ from the underlying one. This is achieved by leveraging PancakeSwap to exchange these tokens for the required asset before initiating the `supply` or `repay` operations. Alongside the main SwapRouter contract, a collection of supporting libraries is used to conduct necessary checks for appropriate interactions with PancakeSwap V2 pools.

Security Model and Trust Assumptions

Venus SwapRouter interacts with PancakeSwap V2 pools and Venus markets, operating under the assumption that these protocols function as intended according to their respective contract specifications.

The contract does not hold user funds, however it may have user approvals.

Privileged Roles

There is only one privileged role which can change the vBNB address and transfer mistakenly sent ERC-20 tokens out of the SwapRouter contract.

Low Severity

L-01 Missing Docstrings

Throughout the codebase, there are several parts that do not have docstrings. For example:

- Line 18 in `IRouterHelper.sol`
- Line 20 in `IRouterHelper.sol`
- Line 14 in `RouterHelper.sol`
- Line 269 in `RouterHelper.sol`

Consider thoroughly documenting all functions (and their parameters) that are part of any contract's public API. Functions implementing sensitive functionality, even if not public, should be clearly documented as well. When writing docstrings, consider following the Ethereum Natural Specification Format (NatSpec).

Update: *Resolved in pull request #281 at commits 394d1a7 and 9751c85. The functions were documented in their definitions in the contracts instead of their declarations in the interfaces.*

L-02 Locked BNB in Contract

There are multiple occurrences in `SwapRouter.sol` where ETH can become locked. For instance:

- `swapExactTokensForBNBAndRepay`
- `swapExactTokensForBNBAndRepayAtSupportingFee`
- `swapTokensForExactBNBAndRepay`
- `swapTokensForFullBNBDebtAndRepay`

Consider removing the `payable` attribute or adding a withdrawal feature.

Update: *Resolved in pull request #281 at commit 7a8044a.*

Notes & Additional Information

N-01 Misleading Docstrings

Line 811 in `SwapRouter.sol` states that the function checks if the value is "greater than" when it actually checks if the value is "greater or equal to".

Consider correcting this docstring.

Update: *Resolved in pull request #281 at commit 20e3118.*

N-02 Naming Can Be Improved

- In line 16 of `CustomErrors.sol`, the second parameter is named `currentBlock` even though the value actually contains `timestamp`.
- In line 29 of `TransferHelper.sol`, `TransferFromFailed` can be renamed to `SafeTransferFromFailed`, making it consistent with other similar error names.
- In lines 269 and 293 of `SwapRouter.sol`, `swapAndRepay` and `swapAndRepayAtSupportingFee` can be renamed to `swapExactTokensForTokensAndRepay` and `swapExactTokensForTokensAndRepayAtSupportingFee`, making them consistent with other similar function names.

Consider amending the aforementioned names for clarity and consistency.

Update: *Resolved in pull request #281 at commit 027835e.*

N-03 Some Convenience Functions Are Missing

The `SwapRouter` contract has functions to facilitate `supply` and `repay` operations for ERC-20 tokens. However, some of the functions are missing.

Here are some examples of missing functions, although this is not an exhaustive list.

- `swapExactTokensForBNBAndSupply`
- `swapExactTokensForBNBAndSupplyAtSupportingFee`

- `swapTokensForExactBNBAndSupply`
- `swapTokensForExactBNBAndSupplyAtSupportingFee`

Consider adding them to offer more convenience functions to the users.

Update: *Resolved in pull request #281 at commits fb66414 and cf6b8cb. All the suggested functions were implemented with the exception of* `swapTokensForExactBNBAndSupplyAtSupportingFee`*. The Venus team stated:*

> *New functions added to the SwapRouter contract:*
>
> - `swapExactTokensForBNBAndSupply`
> - `swapExactTokensForBNBAndSupplyAtSupportingFee`
> - `swapTokensForExactBNBAndSupply`
>
> *We cannot implement* `swapTokensForExactBNBAndSupplyAtSupportingFee` *as* `swapTokensForExactBNBAtSupportingFee` *does not exist in the* `RouterHelper` *because getting the amount of the tokens (with fees on transfer) to swap for an exact amount of BNB is not possible.*

N-04 Confusing Use of ETH and BNB in Comments and Function Names

BNB and ETH are referenced multiple times interchangeably throughout the codebase. For example, in lines 129, 154, 221, and 241. The actual deployment will be on BSC and ETH mentions are a result of PancakeSwap being a Uniswap fork.

To increase the clarity of the codebase, consider changing all mentions of ETH to BNB.

Update: *Resolved in pull request #281 at commits bbe298f, 9751c85, and dbf855c.*

N-05 Lack of SPDX License Identifiers

Throughout the codebase, there are files that lack SPDX license identifiers. For instance:

- `IRouterHelper.sol`
- `SwapRouter.sol`
- `CustomErrors.sol`
- `IPancakePair.sol`
- `IPancakeSwapV2Factory.sol`

- `IPancakeSwapV2Router.sol`
- `IVBNB.sol`
- `IVtoken.sol`
- `InterfaceComptroller.sol`
- `PancakeLibrary.sol`

To avoid legal issues regarding copyright and follow best practices, consider adding SPDX license identifiers to files as suggested by the Solidity documentation.

Update: *Resolved in pull request #281 at commit 8b08294.*

N-06 `PancakeRouter` Functions' Code Can Be Reused

The `SwapRouter` contract adapts the swap router logic from the `PancakeRouter` contract. It also adds code that operates either before or after a swap. The adapted code increases the complexity of the `SwapRouter` contract which makes it more prone to potential bugs.

Consider calling the `PancakeRouter` functions instead of adapting their code to the `SwapRouter` contract.

Update: *Acknowledged, not resolved. The Venus team stated:*

> *Our `PancakeRouter` contract uses a newer Solidity version and reverts with custom errors instead of `require` statements. We prefer to maintain and use our version.*

Conclusions

The system implements all common security checks for swap routers, demonstrating adherence to industry standards. However, some changes have been proposed to follow best practices and reduce the potential attack surface.

Appendix

Monitoring Recommendations

While the audit helps identify code-level issues in the current implementation, the Venus team is encouraged to incorporate monitoring activities for deployed contracts. Specifically, it is recommended to set up an alert system for when funds are mistakenly sent to the contract in order to facilitate prompt recovery. In addition, monitor the `setVBNBAdress` function that implements the `onlyOwner` modifier to ensure that all admin actions are authorized by the team and that the VBNB address remains correct.