BELT FINANCE SECURITY ASSESSMENT REPORT

MAY. 31 ~ JUN. 15, 2021

DISCLAIMER

• This document is based on a security assessment conducted by a blockchain security company SOOHO. This document describes the detected security vulnerabilities and also discusses the code quality and code license violations.

• This security assessment does not guarantee nor describe the usefulness of the code, the stability of the code, the suitability of the business model, the legal regulation of the business, the suitability of the contract, and the bug-free status. Audit document is used for discussion purposes only.

• SOOHO does not disclose any business information obtained during the review or save it through a separate media.

• SOOHO presents its best endeavors in smart contract security assessment.

SOOHO

SOOHO with the motto of “Audit Everything, Automatically” researches and provides technology for reliable blockchain ecosystem. SOOHO verifies vulnerabilities through entire development life-cycle with Aegis, a vulnerability analyzer created by SOOHO, and open source analyzers. SOOHO is composed of experts including Ph.D researchers in the field of automated security tools and white-hackers verifying contract codes and detected vulnerabilities in depth. Professional experts in SOOHO secure partners’ contracts from known to zero-day vulnerabilities.

INTRODUCTION

SOOHO conducted a security assessment of Ozys's BELT Heco smart contract from May. 31 until Jun. 15. The following tasks were performed during the audit period:

• Performing and analyzing the results of Odin, a static analyzer of SOOHO.

• Writing Exploit codes on suspected vulnerability in the contract.

• Recommendations on codes based on best practices and the Secure Coding Guide.

A total of three security experts participated in a vulnerability analysis of the contract. The experts are professional hackers with Ph.D. academic backgrounds and experiences of receiving awards from national/international hacking competitions such as Defcon, Nuit du Hack, White Hat, SamsungCTF, and etc.

The detected vulnerabilities are as follows: Low 1, Note 1. It is recommended to promote the stability of service through continuous code audit and analyze potential vulnerabilities.

ANALYSIS TARGET

The following projects were analyzed during period.

Project	belt-contract-latest/contracts/heco
Commit #	d08d73d1
# of Files	35
# of Lines	4,671

KEY AUDIT POINTS & PROCESS

BELT Heco is an AMM protocol that incorporates multi-strategy yield optimizing on Huobi Eco Chain (Heco). BELT Heco contains 3 different strategy which connected with lending protocol. Accordingly, we mainly reviewed common vulnerabilities in DeFi services and possible hacking scenarios.

For example, the following scenarios are included: draining the contract's funds, freezing funds, breaking pools. However, we did not take any internal hackings by administrators into account. Additionally, although not mentioned in this report, we would like to suggest the customer's interest in the stability of external services as well. Most analyzes are about the functioning of the subject contract, given the safety of the system.

1

Automated Vulnerability Analysis

2

Manual Code Analysis

The followings are considered:

- Preferential analysis of codes with greater risks.
- Supervision of Access Control management.
- Analyze whether the code is written under the client's intention.

3

Review of Exploitability and PoC Code

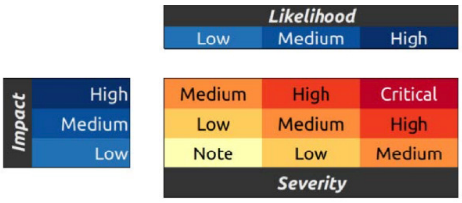
The followings are considered:

- Dynamic analysis through code execution.
- Examine possible financial gain by misusing detected vulnerabilities. (e.g., infinite withdrawal)
- Examine the possibility of adverse effect to the token service by misusing detected vulnerabilities. (e.g., Mint)

RISK RATING OF VULNERABILITY

Detected vulnerabilities are listed on the basis of the risk rating of vulnerability.

The risk rating of vulnerability is set based on OWASP's Impact & Likelihood Risk Rating Methodology as seen on the right. Some issues were rated vulnerable aside from the corresponding model and the reasons are explained in the following results.



Impact \ Likelihood	Low	Medium	High
High	Medium	High	Critical
Medium	Low	Medium	High
Low	Note	Low	Medium

Severity

ANALYSIS RESULTS

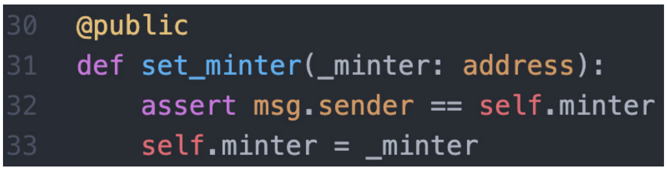
Analysis results are categorized into Critical, High, Medium, Low, and Note. SOOHO recommends upgrades on every detected issue.

MINTER CAN BE NULL Low

Additional resources and comments

File Name : `BeltLPTokenv2.sol`

File Location : `belt-contract/../heco/swapV2`
`└── BeltLPTokenv2.vy`



```
@public
def set_minter(_minter: address):
    assert msg.sender == self.minter
    self.minter = _minter
```

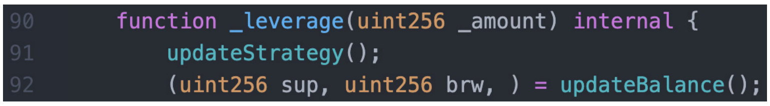
Details `_minter` needs to be validated. Also, we recommend to emit Events for broadcaset the changes.

INTEREST RATES MAY OUTDATED Note

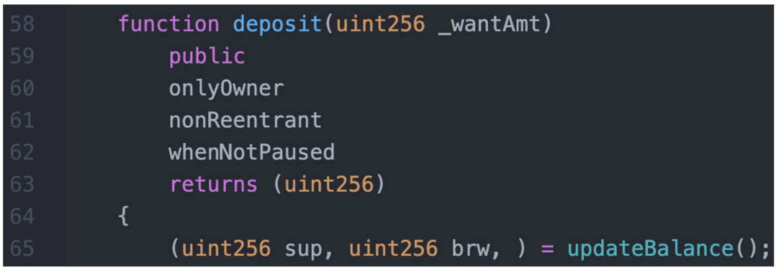
Additional resources and comments

File Name : `Strategy**Impl.sol`

File Location : `belt-contract/../heco/earnV2/strategies/**/`
`└── Strategy**Impl.sol`



```
    function _leverage(uint256 _amount) internal {
        updateStrategy();
        (uint256 sup, uint256 brw, ) = updateBalance();
```



```
    function deposit(uint256 _wantAmt)
        public
        onlyOwner
        nonReentrant
        whenNotPaused
        returns (uint256)
    {
        (uint256 sup, uint256 brw, ) = updateBalance();
```

Details `updateStrategy` function invokes CToken's `accrueInterest` function to add interest for suppliers and borrowers in the market. Also, it will add after executing mint, redeem, borrow, repay function. Therefore, `accrueInterest` need to be executed before invoking `updateBalance`. Accordinly, `mint` and `withdraw` function need to call `updateStrategy` for the latest balance.

ADDITIONAL ANALYSIS RESULTS

Additional analysis results include a description of the main areas we looked at during the analysis.

ARITHMETIC ROUNDING ✓

Details Abusing through arithmetic rounding and fund drain were analyzed, but none were found.

Additional resources and comments

ARITHMETIC ISSUES ✓

Details Arithmetic operations are using SafeMath

Additional resources and comments

CONCLUSION

The source code of the BELT Finance is easy to read and very well organized. We have to remark that contracts are well architected and all the additional features are implemented. **The detected vulnerabilities are as follows: Low 1, Note 1.** However, most of the codes are found out to be compliant with all the best practices. It is recommended to promote the stability of service through continuous code audit and analyze potential vulnerabilities.

Project	belt-contract/contracts/heco
Commit #	d08d73d1
# of Files	35
# of Lines	4,671

File Tree

```
belt-contract/contracts/heco
├── Migrations.sol
├── earnV2
│   ├── MasterOrbit.sol
│   ├── defi
│   │   ├── channels.sol
│   │   ├── filda.sol
│   │   ├── lendHub.sol
│   │   └── mdex.sol
│   ├── strategies
│   │   ├── Strategy.sol
│   │   ├── channels
│   │   │   ├── StrategyChannels.sol
│   │   │   ├── StrategyChannelsImpl.sol Note
│   │   │   └── StrategyChannelsStorage.sol
│   │   ├── filda
│   │   │   ├── StrategyFilda.sol
│   │   │   ├── StrategyFildaImpl.sol Note
│   │   │   └── StrategyFildaStorage.sol
│   │   └── lendhub
│   │       ├── StrategyLendHub.sol
│   │       ├── StrategyLendHubImpl.sol Note
│   │       └── StrategyLendHubStorage.sol
│   └── tokens
│       ├── MultiStrategyToken.sol
│       ├── MultiStrategyTokenImpl.sol
│       ├── MultiStrategyTokenStorage.sol
│       ├── SingleStrategyToken.sol
│       ├── SingleStrategyToken2.sol
│       ├── SingleStrategyTokenImpl.sol
│       ├── SingleStrategyTokenImpl2.sol
│       ├── SingleStrategyTokenStorage.sol
│       └── StrategyToken.sol
├── swapV2
│   ├── BeltLPTokenv2.vy Low
│   ├── BuyBack4BELT.HECO.sol
│   ├── DepositBv2.vy
│   └── StableSwapBv2.vy
├── utils
│   ├── Timelock.sol
│   └── UnwrapperHT.sol
└── view
    ├── BeltSwapView.sol
    ├── BeltVaultView.sol
    ├── BeltView.sol
    └── TokenPriceView.sol
```