BELT FINANCE SECURITY ASSESSMENT REPORT

MAY. 31 ~ JUN. 15, 2021

시작하기 전에

- 본 문서는 블록체인 보안 전문업체 SOOHO에서 진행한 취약점 검사를 바탕으로 작성한 문서로, 보안 취약점의 발견에 초점을 두고 있습니다. 추가적으로 코드 퀄리티 및 코드 라이센스 위반 사항 등에 대해서도 논의합니다.
- 본 문서는 코드의 유용성, 코드의 안정성, 비즈니스 모델의 적합성, 비즈니스의 법적인 규제, 계약의 적합성, 버그 없는 상태에 대해 보장하거나 서술하지 않습니다. 감사 문서는 논의 목적으로만 사용됩니다.
- SOOHO는 회사 정보가 대외비 이상의 성격을 가짐을 인지하고 사전 승인 없이 이를 공개하지 않습니다.
- SOOHO는 업무 수행 과정에서 취득한 일체의 회사 정보를 누설하거나 별도의 매체를 통해 소장하지 않습니다.
- SOOHO는 스마트 컨트랙 분석에 최선을 다하였음을 밝히는 바입니다.

SOOHO 소개

SOOHO는 Audit Everything, Automatically란 슬로건으로 지속적인 보안을 위해 필요한 기술을 연구하고 서비스합니다. 자체 취약점 분석기들과 오픈소스 분석기들을 기반으로 모든 개발 생애 주기에 걸쳐 취약점들을 검사합니다. SOOHO는 자동화 도구를 연구, 개발하는 보안 분야 박사 연구원들과 탐지 결과와 컨트랙트 코드를 깊게 분석하는 화이트 해커들로 구성되어 있습니다. 보안 분야 전문성을 바탕으로 파트너 사의 컨트랙트를 알려진 취약점과 Zero-day 취약점의 위협으로부터 안전하게 만들어줍니다.

개 요

2021년 5월 31일부터 6월 15일까지 오지스의 BELT Finance Heco 컨트랙트의 대한 취약점 분석을 진행하였습니다. 감사 기간 동안 아래의 작업을 수행했습니다.

- SOOHO의 자체 취약점 검사기를 통한 취약점 탐지 및 결과 분석
- 보안 취약점 의심 지점에 대한 익스플로잇(Exploit) 코드 작성
- 컨트랙트 코드 모범 사례와 시큐어 코딩 가이드를 바탕으로 코드의 수정 권고 사항 작성

보안 전문가들이 컨트랙트의 취약점을 분석하였습니다. 참여한 보안 전문가는 국내외의 블록체인 해커톤 해킹 대회에서 수상을 하고 보안분야 박사 학위의 학문적 배경을 가지는 등 우수한 해킹 실력과 경험을 가지고 있습니다.

분석 결과 **이슈는 총 2개로 심각도 순서대로 Low 1, Note 1개입니다.** 꾸준한 코드 감사를 통해 서비스의 안정을 도모하고 잠재적인 취약점에 대한 분석을 하는 것을 추천 드립니다.

분석 대상

분석 기간 동안 아래의 프로젝트를 분석하였습니다.

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

주요 감사 포인트 및 프로세스

BELT Finance의 Huobi ECO Chain (HECO)에서 동작하는 최적화된 수익률을 낮은 수수료로 제공하는 AMM 프로토콜 입니다. 3가지 대출 서비스 프로토콜과 연계된 기능을 제공합니다. 따라서, DeFi 환경에서 주로 발생 가능한 이슈와 고려되어야 하는 사항들에 대한 분석이 진행되었습니다.

예를 들어, 계약 자금이 소진되고 자금이 동결되고 풀이 중단되는 것을 포함합니다. 단, 관리자의 내부 해킹은 고려하지 않았습니다. 또한 본 보고서에서는 언급하지 않았지만 외부 서비스의 안정성에 대해서도 검토하기를 제안합니다. 분석은 대상 프로젝트에 포함된 컨트랙트의 기능 안정성에 관한 것입니다.

1

취약점 자동 분석
소프트웨어 검사

2

보안 전문가의
코드 직접 분석

다음을 고려합니다.

- 보다 큰 리스크를 내포하는 코드를 중점적으로 분석합니다.
- 접근 권한의 관리가 잘 되고 있는지 검사합니다.
- 파트너 사가 의도한대로 코드가 작성되었는지 분석합니다.

3

Exploit 분석과
Poc 작성

Exploit 분석과 PoC 작성은 다음을 목표로 합니다.

- 코드 실행을 통한 동적 분석
- 발견된 취약점을 악용해 서비스에 악영향을 끼칠 수 있는지 살펴봅니다.

취약점의 심각성 척도

발견된 취약점은 심각성 척도를 기준으로 나열해서 설명합니다.

Critical High Medium Low Note

심각성 척도는 우측 OWASP의 Impact & Likelihood 기반 리스크 평가 모델을 기반으로 정해졌습니다. 해당 모델과 별개로 심각도가 부여된 이슈는 해당 결과에서 그 이유를 서술합니다.

Impact	Likelihood: Low	Likelihood: Medium	Likelihood: High
High	Medium	High	Critical
Medium	Low	Medium	High
Low	Note	Low	Medium

Severity

분석 결과

분석 결과는 심각도에 따라 Critical, High, Medium, Low, Note로 표현됩니다. Sooho는 발견된 모든 이슈에 대해서 개선하는 것을 권장합니다.

MINTER CAN BE NULL Low

분석 결과에 대한 추가적인 자료 및 코멘트

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

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

이슈 설명 `_minter`에 대한 null 체크가 필요합니다. 또한 minter가 변경될 때는 event 함수를 호출하는 것을 권장합니다.

INTEREST RATES MAY OUTDATED Note

분석 결과에 대한 추가적인 자료 및 코멘트

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

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

```
58      function deposit(uint256 _wantAmt)
59          public
60          onlyOwner
61          nonReentrant
62          whenNotPaused
63          returns (uint256)
64      {
65          (uint256 sup, uint256 brw, ) = updateBalance();
```

이슈 설명 `updateStrategy` 함수는 CToken의 `accrueInterest` 함수를 호출하여 마켓의 모든 supplier와 borrower의 이자를 추가합니다. 또한, CToken의 mint, redeem, borrow, repay 함수가 호출된 이후에는 마찬가지 해당 함수가 호출됩니다. 이에 따라 CToken의 스냅샷을 가져오는 `updateBalance` 호출하기 전 해당 함수가 호출되야 가장 최신값을 조회할 수 있습니다. 따라서 Impl 컨트랙트 상의 `mint`와 `withdraw` 함수에서 `updateStrategy`를 실행하는 것을 권장합니다.

분석 결과

분석 결과는 심각도에 따라 Critical, High, Medium, Low, Note로 표현됩니다. Sooho는 발견된 모든 이슈에 대해서 개선하는 것을 권장합니다.

ARITHMETIC ROUNDING ✓

분석 결과에 대한 추가적인 자료 및 코멘트

설명 산술 연산에서의 반올림과 그로 인한 자금 손실을 분석했지만 발견되지 않았습니다.

ARITHMETIC ISSUE ✓

분석 결과에 대한 추가적인 자료 및 코멘트

설명 산술 연산은 모두 안전한 연산으로 처리되었습니다.

검사 결과 요약 및 결론

오지스에서 개발한 BELT Finance 컨트랙트는 이해하기 쉽게 명명되고 용도와 쓰임에 따라 잘 설계되어 있습니다. 대부분 모범 사례를 따르고 있습니다. 코드 검사 결과, **이슈는 총 1개로 심각도 순서대로 Note 1개입니다**. 꾸준한 코드 감사를 통해 컨트랙트의 안정을 도모하고 잠재적인 취약점에 대한 분석을 하는 것을 추천드립니다.

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
```