Smart Contract
Security Audit Report

Table Of Contents

1 Executive Summary

On 2023.08.21, the SlowMist security team received the Helio Money team's security audit application for helio-smart-contracts-eth-collateral, developed the audit plan according to the agreement of both parties and the characteristics of the project, and finally issued the security audit report.

The SlowMist security team adopts the strategy of "white box lead, black, grey box assists" to conduct a complete security test on the project in the way closest to the real attack.

The test method information:

Test method	Description
Black box testing	Conduct security tests from an attacker's perspective externally.
Grey box testing	Conduct security testing on code modules through the scripting tool, observing the internal running status, mining weaknesses.
White box testing	Based on the open source code, non-open source code, to detect whether there are vulnerabilities in programs such as nodes, SDK, etc.

The vulnerability severity level information:

Level	Description
Critical	Critical severity vulnerabilities will have a significant impact on the security of the DeFi project, and it is strongly recommended to fix the critical vulnerabilities.
High	High severity vulnerabilities will affect the normal operation of the DeFi project. It is strongly recommended to fix high-risk vulnerabilities.
Medium	Medium severity vulnerability will affect the operation of the DeFi project. It is recommended to fix medium-risk vulnerabilities.
Low	Low severity vulnerabilities may affect the operation of the DeFi project in certain scenarios. It is suggested that the project team should evaluate and consider whether these vulnerabilities need to be fixed.
Weakness	There are safety risks theoretically, but it is extremely difficult to reproduce in engineering.
Suggestion	There are better practices for coding or architecture.

2 Audit Methodology

The security audit process of SlowMist security team for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the smart contract:

Serial Number	Audit Class	Audit Subclass
1	Overflow Audit	-
2	Reentrancy Attack Audit	-
3	Replay Attack Audit	-
4	Flashloan Attack Audit	-
5	Race Conditions Audit	Reordering Attack Audit
6	Permission Vulnerability Audit	Access Control Audit
		Excessive Authority Audit
7	Security Design Audit	External Module Safe Use Audit
		Compiler Version Security Audit
		Hard-coded Address Security Audit
		Fallback Function Safe Use Audit
		Show Coding Security Audit
		Function Return Value Security Audit
		External Call Function Security Audit

Serial Number	Audit Class	Audit Subclass
7	Security Design Audit	Block data Dependence Security Audit
		tx.origin Authentication Security Audit
8	Denial of Service Audit	-
9	Gas Optimization Audit	-
10	Design Logic Audit	-
11	Variable Coverage Vulnerability Audit	-
12	"False Top-up" Vulnerability Audit	-
13	Scoping and Declarations Audit	-
14	Malicious Event Log Audit	-
15	Arithmetic Accuracy Deviation Audit	-
16	Uninitialized Storage Pointer Audit	-

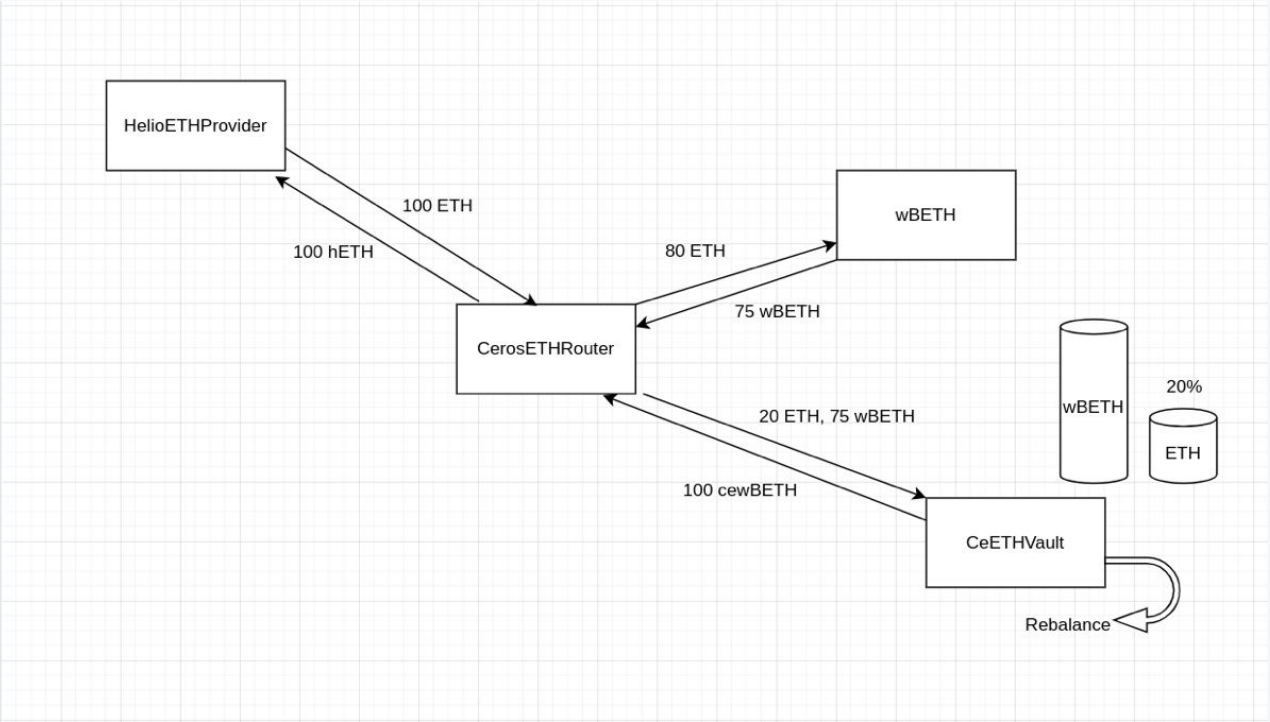
3 Project Overview

3.1 Project Introduction

Helio Protocol is USD decentralized stablecoin backed by BNB.

This audit primarily focuses on incremental code review, with the main addition being the collateralization functionality for BETH.

Convert BETH to hETH flow chart:



3.2 Vulnerability Information

The following is the status of the vulnerabilities found in this audit:

NO	Title	Category	Level	Status
N1	Missing check ERC20 interface return value	Design Logic Audit	Medium	Fixed
N2	Missing zero address validation	Design Logic Audit	Low	Acknowledged
N3	Missing check EthOracle status	Design Logic Audit	High	Fixed
N4	There is a risk of BETH/WBETH deviating from the price anchor of ETH.	Unsafe External Call Audit	Suggestion	Acknowledged
N5	Risk of excessive authority	Authority Control Vulnerability Audit	Medium	Acknowledged
N6	Lack of modifiers	Authority Control Vulnerability Audit	Low	Acknowledged
N7	External call `_BETH.exchangeRate()` risk	Unsafe External Call Audit	Suggestion	Acknowledged

NO	Title	Category	Level	Status
N8	Initializers could be front-run	Reordering Vulnerability	Low	Acknowledged
N9	Missing check return value	Design Logic Audit	Low	Acknowledged

4 Code Overview

4.1 Contracts Description

https://github.com/helio-money/helio-smart-contracts/tree/eth-collateral

Initial audit commit: c0be55ce6762a76818935154cdfc2bb9e0275c4f

Final audit version: 76bbcae678164b83ce1ac25c21c59c65b03b2cd8

The main network address of the contract is as follows:

The code was not deployed to the mainnet.

4.2 Visibility Description

The SlowMist Security team analyzed the visibility of major contracts during the audit, the result as follows:

CeETHVault			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
depositFor	External	Can Modify State	nonReentrant onlyRouter
_deposit	Private	Can Modify State	-
claimYieldsFor	External	Can Modify State	onlyRouter nonReentrant
_claimYields	Private	Can Modify State	-
withdrawETHFor	External	Can Modify State	nonReentrant onlyRouter
_withdrawETH	Private	Can Modify State	-

SLOWMIST Focusing on Blockchain Ecosystem Security

CeETHVault			
withdrawBETHFor	External	Can Modify State	nonReentrant onlyRouter
_withdrawBETH	Private	Can Modify State	-
rebalance	External	Can Modify State	onlyStrategist
getTotalBETHAmountInVault	External	-	-
getTotalETHAmountInVault	External	-	-
getPrincipalOf	External	-	-
getYieldFor	External	-	-
getCeTokenBalanceOf	External	-	-
getDepositOf	External	-	-
getClaimedOf	External	-	-
changeRouter	External	Can Modify State	onlyOwner
changeWithdrawalFee	External	Can Modify State	onlyOwner
setStrategist	External	Can Modify State	onlyOwner
changeCertToken	External	Can Modify State	onlyOwner
getName	External	-	-
getCeToken	External	-	-
getBETHAddress	External	-	-
getRouter	External	-	-

CerosETHRouter			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
deposit	External	Can Modify State	onlyProvider nonReentrant
claim	External	Can Modify State	nonReentrant
withdrawETH	External	Can Modify State	onlyProvider nonReentrant
withdrawBETH	External	Can Modify State	onlyProvider nonReentrant

CerosETHRouter			
liquidation	External	Can Modify State	onlyProvider nonReentrant
changeVault	External	Can Modify State	onlyOwner
changeProvider	External	Can Modify State	onlyOwner
changeMinStakeAmount	External	Can Modify State	onlyOwner
changeCertTokenRatio	External	Can Modify State	onlyOwner
getProvider	External	-	-
getCeToken	External	-	-
getCertToken	External	-	-
getCertTokenRatio	External	-	-
getReferral	External	-	-
getVaultAddress	External	-	-
getMinStake	External	-	-

HelioETHProvider			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
provideInETH	External	Can Modify State	whenNotPaused nonReentrant
claim	External	Can Modify State	nonReentrant onlyOperator
releaseInBETH	External	Can Modify State	whenNotPaused nonReentrant
releaseInETH	External	Can Modify State	whenNotPaused nonReentrant
liquidation	External	Can Modify State	onlyProxy nonReentrant
daoBurn	External	Can Modify State	onlyProxy nonReentrant
daoMint	External	Can Modify State	onlyProxy nonReentrant

HelioETHProvider			
_provideCollateral	Internal	Can Modify State	-
_withdrawCollateral	Internal	Can Modify State	-
pause	External	Can Modify State	onlyOwner
unPause	External	Can Modify State	onlyOwner
changeDao	External	Can Modify State	onlyOwner
changeCeToken	External	Can Modify State	onlyOwner
changeProxy	External	Can Modify State	onlyOwner
changeCollateralToken	External	Can Modify State	onlyOwner
changeOperator	External	Can Modify State	onlyOwner
changeCertToken	External	Can Modify State	onlyOwner
changeMinWithdrwalAmount	External	Can Modify State	onlyOwner

hETH			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
burn	External	Can Modify State	onlyMinter
mint	External	Can Modify State	onlyMinter
changeMinter	External	Can Modify State	onlyOwner
getMinter	External	-	-

4.3 Vulnerability Summary

[N1] [Medium] Missing check ERC20 interface return value

Category: Design Logic Audit

Content

Code location:

```
_BETH.transferFrom(msg.sender,address(this),wBETHAmount)
(ceros/ETH/CeETHVault.sol#77)
_certToken.transferFrom(msg.sender,address(this),certTokenAmount)
(ceros/ETH/CeETHVault.sol#78)
_BETH.transfer(recipient,availableYields) (ceros/ETH/CeETHVault.sol#106)
_BETH.transfer(recipient,balance) (ceros/ETH/CeETHVault.sol#109)
_certToken.transfer(recipient,amountInETH) (ceros/ETH/CeETHVault.sol#110)
_certToken.transfer(recipient,amount - feeCharged) (ceros/ETH/CeETHVault.sol#138)
_certToken.transfer(referral,feeCharged) (ceros/ETH/CeETHVault.sol#140)
_BETH.transfer(recipient,realAmount) (ceros/ETH/CeETHVault.sol#169)

_certToken.approve(address(_vault),0) (ceros/ETH/CerosETHRouter.sol#156)
_certToken.approve(address(_vault),type()(uint256).max)
(ceros/ETH/CerosETHRouter.sol#158)
```

Solution

Check return value, or use `safeTransfer/safeTransferFrom/safeApprove`.

Status

Fixed

[N2] [Low] Missing zero address validation

Category: Design Logic Audit

Content

```
CeETHVault.initialize(string,address,address,address,uint256,address).strategist
(ceros/ETH/CeETHVault.sol#49) lacks a zero-check on :
                - _strategist = strategist (ceros/ETH/CeETHVault.sol#59)
CeETHVault.changeRouter(address).router (ceros/ETH/CeETHVault.sol#239) lacks a zero-
check on :
                - _router = router (ceros/ETH/CeETHVault.sol#240)
CeETHVault.setStrategist(address).strategist (ceros/ETH/CeETHVault.sol#247) lacks a
zero-check on :
                - _strategist = strategist (ceros/ETH/CeETHVault.sol#248)

CerosETHRouter.initialize(address,address,address,address,uint256,address,uint256).re
ferral (ceros/ETH/CerosETHRouter.sol#48) lacks a zero-check on :
```

```
                - _referral = referral (ceros/ETH/CerosETHRouter.sol#58)
CerosETHRouter.changeProvider(address).provider (ceros/ETH/CerosETHRouter.sol#161)
lacks a zero-check on :
                - _provider = provider (ceros/ETH/CerosETHRouter.sol#162)

HelioETHProvider.initialize(address,address,address,address,address,uint256).certToke
n (ceros/ETH/HelioETHProvider.sol#51) lacks a zero-check on :
                - _certToken = certToken (ceros/ETH/HelioETHProvider.sol#62)
HelioETHProvider.initialize(address,address,address,address,address,uint256).ceToken
(ceros/ETH/HelioETHProvider.sol#52) lacks a zero-check on :
                - _ceToken = ceToken (ceros/ETH/HelioETHProvider.sol#63)
HelioETHProvider.changeCeToken(address).ceToken (ceros/ETH/HelioETHProvider.sol#189)
lacks a zero-check on :
                - _ceToken = ceToken (ceros/ETH/HelioETHProvider.sol#191)
HelioETHProvider.changeProxy(address).auctionProxy
(ceros/ETH/HelioETHProvider.sol#195) lacks a zero-check on :
                - _proxy = auctionProxy (ceros/ETH/HelioETHProvider.sol#196)
HelioETHProvider.changeOperator(address).operator
(ceros/ETH/HelioETHProvider.sol#203) lacks a zero-check on :
                - _operator = operator (ceros/ETH/HelioETHProvider.sol#204)
HelioETHProvider.changeCertToken(address).token (ceros/ETH/HelioETHProvider.sol#207)
lacks a zero-check on :
                - _certToken = token (ceros/ETH/HelioETHProvider.sol#208)

hETH.changeMinter(address).minter (ceros/ETH/hETH.sol#42) lacks a zero-check on :
                - _minter = minter (ceros/ETH/hETH.sol#43)
```

Solution

Check that the address is not zero.

Status

Acknowledged

[N3] [High] Missing check EthOracle status

Category: Design Logic Audit

Content

- contracts/oracle/EthOracle.sol

```
function peek() public view returns (bytes32, bool) {
    (
        /*uint80 roundID*/,
        int price,
```

```
            /*uint startedAt*/,
            /*uint timeStamp*/,
            /*uint80 answeredInRound*/
        ) = priceFeed.latestRoundData();
        if (price < 0) {
            return (0, false);
        }
        return (bytes32(uint(price) * (10**10)), true);
    }
```

The `peek()` function in the contract fetches price data from `priceFeed`, but it does not perform validity checks on the retrieved timestamp `timeStamp`. Consequently, if the fetched price data is outdated, the function might return inaccurate results.

Solution

In the `peek()` function, it is advised to add a validity check for the timestamp (`timeStamp`). You can examine whether the timestamp falls within a reasonable range to ensure that the fetched price data is up-to-date.

Status

Fixed

[N4] [Suggestion] There is a risk of BETH/WBETH deviating from the price anchor of ETH.

Category: Unsafe External Call Audit

Content

Wrapped Beacon ETH (WBETH) is a new liquid staking token, where 1 WBETH represents 1 BETH. WBETH is primarily utilized for staking and mining, yet it carries a risk of slashing. In the event of significant slashing, there's a potential for WBETH to deviate from its pegged price, thus leading to liquidation risks.

Solution

Avoid excessive reliance on the value of WBETH.

Status

Acknowledged

[N5] [Medium] Risk of excessive authority

Category: Authority Control Vulnerability Audit

Content

Owner or special administrator accounts can operate the key functions.

```
HelioETHProvider changeDao
HelioETHProvider changeCeToken
HelioETHProvider changeProxy
HelioETHProvider changeCollateralToken
HelioETHProvider changeOperator
HelioETHProvider changeCertToken
HelioETHProvider changeMinWithdrwalAmount

CerosETHRouter   changeVault
CerosETHRouter   changeProvider
CerosETHRouter   changeMinStakeAmount
CerosETHRouter   changeCertTokenRatio

CeETHVault       changeRouter
CeETHVault       changeWithdrawalFee
CeETHVault       setStrategist
CeETHVault       changeCertToken

hETH     changeMinter
```

Solution

In the short term, transferring owner ownership to multisig contracts is an effective solution to avoid single-point risk. But in the long run, it is a more reasonable solution to implement a privilege separation strategy and set up multiple privileged roles to manage each privileged function separately. And the authority involving user funds should be managed by the community, and the authority involving emergency contract suspension can be managed by the EOA address. This ensures both a quick response to threats and the safety of user funds.

Status

Acknowledged

[N6] [Low] Lack of modifiers

Category: Authority Control Vulnerability Audit

Content

- contracts/ceros/ETH/HelioETHProvider.sol

```
function claim(address recipient)
external
```

```
override
nonReentrant
onlyOperator
returns (uint256 yields)
{
    yields = _ceETHRouter.claim(recipient);
    emit Claim(recipient, yields);
    return yields;
}
```

Lack of `whenNotPaused`

- contracts/ceros/ETH/CerosETHRouter.sol

```
function claim(address recipient)
external
override
nonReentrant
returns (uint256 yields)
{
    yields = _vault.claimYieldsFor(msg.sender, recipient);
    emit Claim(recipient, address(_certToken), yields);
    return yields;
}
```

Lack of `onlyProvider` or `whenNotPaused`

Solution

Add modifier for the functions.

Status

Acknowledged; Add `onlyProvider`, but not add `whenNotPaused`

[N7] [Suggestion] External call `_BETH.exchangeRate()` risk

Category: Unsafe External Call Audit

Content

```
uint256 BETHAmount = (amount - certTokenAmount) * 1e18 / _BETH.exchangeRate();
(ceros/ETH/CerosETHRouter.sol#85)

uint256 ratio = _BETH.exchangeRate();(ceros/ETH/CeETHVault.sol#76)

uint256 amountInETH = (availableYields - balance) * _BETH.exchangeRate() / 1e18;
```

```
(ceros/ETH/CeETHVault.sol#108)

uint256 ratio = _BETH.exchangeRate();
(ceros/ETH/CeETHVault.sol#156)

uint256 ratio = _BETH.exchangeRate();
(ceros/ETH/CeETHVault.sol#205)
```

The `_BETH.exchangeRate()` determines the exchange ratio between wBETH and cewBETH, and it's a mutable variable controlled by the oracle of the BETH contract. When it changes, it leads to alterations in the exchange ratios of user tokens and impacts the token reserve ratios of the system.

Solution

It's important to vigilantly monitor changes in the `exchangeRate`, as they could potentially expose the system to the risk of insufficient collateral.

Status

Acknowledged

[N8] [Low] Initializers could be front-run

Category: Reordering Vulnerability

Content

- CeETHVault.sol

```
function initialize(
) external initializer {
    __Ownable_init();
    __Pausable_init();
    __ReentrancyGuard_init();
```

- CerosETHRouter.sol

```
function initialize(
) public initializer {
    __Ownable_init();
```

```
    __Pausable_init();
    __ReentrancyGuard_init();
```

- HelioETHProvider.sol

```
function initialize(
) public initializer {
    __Ownable_init();
    __Pausable_init();
    __ReentrancyGuard_init();
```

- hETH.sol

```
function initialize() external initializer {
    __Ownable_init();
```

Initializers could be front-run, allowing an attacker to either set their own values, take ownership of the contract, and in the best case forcing a re-deployment

Solution

Use `constructor` to initialize the contracts.

Status

Acknowledged

[N9] [Low] Missing check return value

Category: Design Logic Audit

Content

Code location:

```
_vault.withdrawETHFor(msg.sender,recipient,amount) (ceros/ETH/CerosETHRouter.sol#147)
_vault.withdrawETHFor(msg.sender,recipient,amount) (ceros/ETH/CerosETHRouter.sol#151)
_vault.withdrawBETHFor(msg.sender,recipient,diff) (ceros/ETH/CerosETHRouter.sol#152)

_dao.deposit(account,address(_ceToken),amount) (ceros/ETH/HelioETHProvider.sol#164)
_dao.withdraw(account,address(_ceToken),amount) (ceros/ETH/HelioETHProvider.sol#168)
```

Solution

Check return value.

Status

Acknowledged

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002308290001	SlowMist Security Team	2023.08.21 - 2023.08.29	Medium Risk

Summary conclusion: The SlowMist security team use a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 1 high risk, 2 medium risk, 4 low risk, 2 suggestion vulnerabilities.

6 Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist