Smart Contract
Security Audit Report

Table Of Contents

1 Executive Summary

On 2023.05.12, the SlowMist security team received the Helio Money team's security audit application for Helio Money, 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

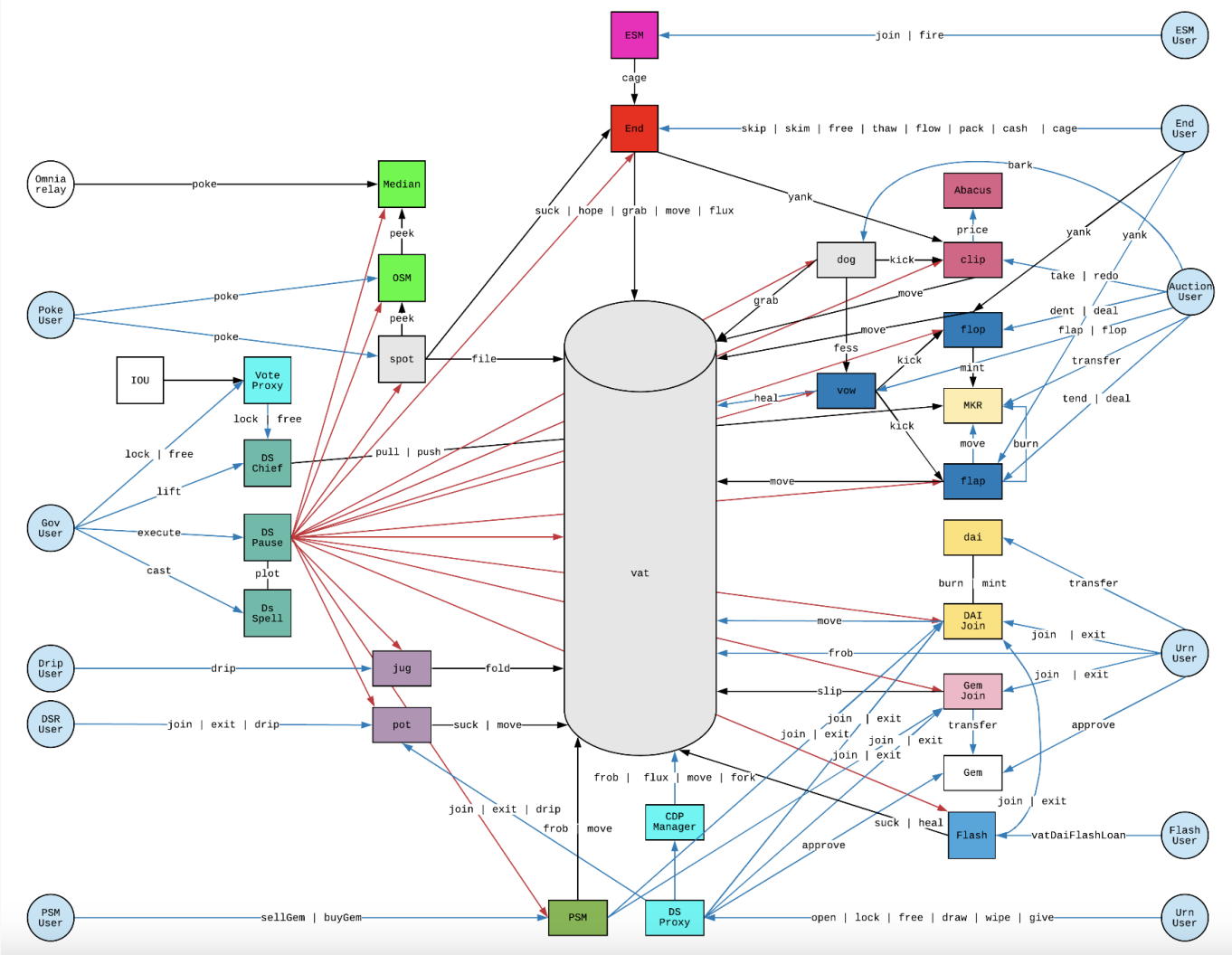
Serial Number	Audit Class	Audit Subclass
7	Security Design Audit	Block data Dependence Security Audit
		tx.origin Authentication Security Audit
8	Denial of Service Audit	-
9	Gas Optimization Audit	-
10	Design Logic Audit	-
11	Variable Coverage Vulnerability Audit	-
12	"False Top-up" Vulnerability Audit	-
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

HAY is an easy-to-use destablecoin on the BNB Chain.

Since Hay has a relatively high similarity to Dai, we can refer to Dai's architecture for analysis:



3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Risk of excessive authority	Authority Control Vulnerability Audit	Medium	Confirmed
N2	Sandwich attacks can affect slippage scope	Reordering Vulnerability	Low	Acknowledged
N3	Ratio arbitrage attack vulnerability	Unsafe External Call Audit	Critical	Fixed
N4	Missing check return value	Design Logic Audit	High	Fixed
N5	`transfer()` and `send()` risks	Denial of Service Vulnerability	Medium	Fixed

NO	Title	Category	Level	Status
N6	Missing zero address validation	Design Logic Audit	Suggestion	Fixed
N7	Performs a multiplication on the result of a division	Arithmetic Accuracy Deviation Vulnerability	Low	Ignored
N8	Anyone can call initialize on the logic contract	Race Conditions Vulnerability	Low	Acknowledged
N9	vaultToken burned may exceed the actual number needed	Design Logic Audit	High	Fixed
N10	`_rewardsToken` can be arbitrary contract	Design Logic Audit	High	Fixed
N11	Missing check BnbOracle status	Unsafe External Call Audit	Medium	Fixed
N12	Upgrading contracts may introduce new risks	Authority Control Vulnerability Audit	Suggestion	Acknowledged
N13	HelioOracle owner is never initialized	Design Logic Audit	High	Fixed
N14	HelioOracle price oracle is not rigorous	Design Logic Audit	Low	Acknowledged
N15	Oracle price should not return 0	Unsafe External Call Audit	Medium	Fixed
N16	ERC777 reentrancy risks	Reentrancy Vulnerability	Medium	Fixed
N17	Missing events access control	Malicious Event Log Audit	Suggestion	Fixed
N18	Reentry prevention best practices	Reentrancy Vulnerability	Suggestion	Fixed
N19	Missing check return value	Design Logic Audit	Low	Acknowledged

4 Code Overview

4.1 Contracts Description

https://github.com/agiledev624/helio-smart-contracts

commit: b8587950d2b1bcc62a8ab3d99198cbae12bc5c78

Review version: 6aa1793448d8fbe78128715e9b8395046ede8d99

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:

Vow			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
min	Internal	-	-
file	External	Can Modify State	auth
file	External	Can Modify State	auth
heal	External	Can Modify State	-
feed	External	Can Modify State	-
flap	External	Can Modify State	-
cage	External	Can Modify State	auth
uncage	External	Can Modify State	auth

LinearDecrease			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
file	External	Can Modify State	auth
add	Internal	-	-
mul	Internal	-	-
rmul	Internal	-	-
price	External	-	-

StairstepExponentialDecrease			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
file	External	Can Modify State	auth
rmul	Internal	-	-
rpow	Internal	-	-
price	External	-	-

ExponentialDecrease			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth

ExponentialDecrease			
initialize	External	Can Modify State	initializer
file	External	Can Modify State	auth
rmul	Internal	-	-
rpow	Internal	-	-
price	External	-	-

Initializable			
Function Name	Visibility	Mutability	Modifiers
_disableInitializers	Internal	Can Modify State	-
_getInitializedVersion	Internal	-	-
_isInitializing	Internal	-	-

HelioToken			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	Public	Can Modify State	initializer
mint	External	Can Modify State	auth
burn	External	Can Modify State	-
pause	External	Can Modify State	auth
unpause	External	Can Modify State	auth

ERC20PausableUpgradeable			
Function Name	Visibility	Mutability	Modifiers
__ERC20Pausable_init	Internal	Can Modify State	onlyInitializing
__ERC20Pausable_init_unchained	Internal	Can Modify State	onlyInitializing
_beforeTokenTransfer	Internal	Can Modify State	-

ERC20Upgradeable			
Function Name	Visibility	Mutability	Modifiers
__ERC20_init	Internal	Can Modify State	onlyInitializing
__ERC20_init_unchained	Internal	Can Modify State	onlyInitializing
name	Public	-	-
symbol	Public	-	-
decimals	Public	-	-
totalSupply	Public	-	-
balanceOf	Public	-	-
transfer	Public	Can Modify State	-
allowance	Public	-	-
approve	Public	Can Modify State	-
transferFrom	Public	Can Modify State	-
increaseAllowance	Public	Can Modify State	-
decreaseAllowance	Public	Can Modify State	-
_transfer	Internal	Can Modify State	-
_mint	Internal	Can Modify State	-
_burn	Internal	Can Modify State	-
_approve	Internal	Can Modify State	-
_spendAllowance	Internal	Can Modify State	-
_beforeTokenTransfer	Internal	Can Modify State	-
_afterTokenTransfer	Internal	Can Modify State	-

ContextUpgradeable			
Function Name	Visibility	Mutability	Modifiers
__Context_init	Internal	Can Modify State	onlyInitializing
__Context_init_unchained	Internal	Can Modify State	onlyInitializing
_msgSender	Internal	-	-
_msgData	Internal	-	-

PausableUpgradeable			
Function Name	Visibility	Mutability	Modifiers
__Pausable_init	Internal	Can Modify State	onlyInitializing
__Pausable_init_unchained	Internal	Can Modify State	onlyInitializing
paused	Public	-	-
_requireNotPaused	Internal	-	-
_requirePaused	Internal	-	-
_pause	Internal	Can Modify State	whenNotPaused
_unpause	Internal	Can Modify State	whenPaused

CerosRouter			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
deposit	External	Payable	nonReentrant
depositABNBcFrom	External	Can Modify State	onlyProvider nonReentrant
depositABNBc	External	Can Modify State	nonReentrant
claim	External	Can Modify State	nonReentrant
claimProfit	External	Can Modify State	nonReentrant

CerosRouter			
withdraw	External	Can Modify State	nonReentrant
withdrawABNBc	External	Can Modify State	nonReentrant
withdrawFor	External	Can Modify State	nonReentrant onlyProvider
withdrawWithSlippage	External	Can Modify State	nonReentrant
getProfitFor	External	-	-
getYieldFor	External	-	-
getPendingWithdrawalOf	External	-	-
changeVault	External	Can Modify State	onlyOwner
changeDex	External	Can Modify State	onlyOwner
changePool	External	Can Modify State	onlyOwner
changeProvider	External	Can Modify State	onlyOwner
getProvider	External	-	-
getCeToken	External	-	-
getWbnbAddress	External	-	-
getCertToken	External	-	-
getPoolAddress	External	-	-
getDexAddress	External	-	-
getVaultAddress	External	-	-

OwnableUpgradeable			
Function Name	Visibility	Mutability	Modifiers
__Ownable_init	Internal	Can Modify State	onlyInitializing
__Ownable_init_unchained	Internal	Can Modify State	onlyInitializing
owner	Public	-	-
_checkOwner	Internal	-	-
renounceOwnership	Public	Can Modify State	onlyOwner

OwnableUpgradeable			
transferOwnership	Public	Can Modify State	onlyOwner
_transferOwnership	Internal	Can Modify State	-

ReentrancyGuardUpgradeable			
Function Name	Visibility	Mutability	Modifiers
__ReentrancyGuard_init	Internal	Can Modify State	onlyInitializing
__ReentrancyGuard_init_unchained	Internal	Can Modify State	onlyInitializing
_nonReentrantBefore	Private	Can Modify State	-
_nonReentrantAfter	Private	Can Modify State	-

CeToken			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
burn	External	Can Modify State	onlyMinter
mint	External	Can Modify State	onlyMinter
changeVault	External	Can Modify State	onlyOwner
getVaultAddress	External	-	-

CeVault			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
deposit	External	Can Modify State	nonReentrant
depositFor	External	Can Modify State	nonReentrant onlyRouter
_deposit	Private	Can Modify State	-

CeVault			
claimYieldsFor	External	Can Modify State	onlyRouter nonReentrant
claimYields	External	Can Modify State	nonReentrant
_claimYields	Private	Can Modify State	-
withdraw	External	Can Modify State	nonReentrant
withdrawFor	External	Can Modify State	nonReentrant onlyRouter
_withdraw	Private	Can Modify State	-
getTotalAmountInVault	External	-	-
getPrincipalOf	External	-	-
getYieldFor	External	-	-
getCeTokenBalanceOf	External	-	-
getDepositOf	External	-	-
getClaimedOf	External	-	-
changeRouter	External	Can Modify State	onlyOwner
getName	External	-	-
getCeToken	External	-	-
getAbnbcAddress	External	-	-
getRouter	External	-	-

hBNB			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
burn	External	Can Modify State	onlyMinter
mint	External	Can Modify State	onlyMinter

hBNB			
changeMinter	External	Can Modify State	onlyOwner
getMinter	External	-	-

NonTransferableERC20			
Function Name	Visibility	Mutability	Modifiers
__ERC20_init	Internal	Can Modify State	onlyInitializing
__ERC20_init_unchained	Internal	Can Modify State	onlyInitializing
name	Public	-	-
symbol	Public	-	-
decimals	Public	-	-
totalSupply	Public	-	-
balanceOf	Public	-	-
transfer	Public	Can Modify State	-
allowance	Public	-	-
approve	Public	Can Modify State	-
transferFrom	Public	Can Modify State	-
_mint	Internal	Can Modify State	-
_burn	Internal	Can Modify State	-
_beforeTokenTransfer	Internal	Can Modify State	-
_afterTokenTransfer	Internal	Can Modify State	-

HelioProvider			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer

HelioProvider			
provide	External	Payable	whenNotPaused nonReentrant
provideInABNBc	External	Can Modify State	nonReentrant
claimInABNBc	External	Can Modify State	nonReentrant onlyOperator
release	External	Can Modify State	whenNotPaused nonReentrant
releaseInABNBc	External	Can Modify State	nonReentrant
liquidation	External	Can Modify State	onlyProxy nonReentrant
daoBurn	External	Can Modify State	onlyProxy nonReentrant
daoMint	External	Can Modify State	onlyProxy nonReentrant
_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

MasterVault			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
depositETH	Public	Payable	nonReentrant whenNotPaused onlyProvider
withdrawETH	External	Can Modify State	nonReentrant whenNotPaused onlyProvider
withdrawFromActiveStrategies	Private	Can Modify State	-
_depositToStrategy	Private	Can Modify State	-

MasterVault			
_updateCerosStrategyDebt	External	Can Modify State	onlyOwner
depositAllToStrategy	Public	Can Modify State	onlyManager
depositToStrategy	Public	Can Modify State	onlyManager
withdrawFromStrategy	Public	Can Modify State	onlyManager
withdrawAllFromStrategy	External	Can Modify State	onlyManager
_withdrawFromStrategy	Private	Can Modify State	-
withdrawInTokenFromStrategy	External	Can Modify State	nonReentrant whenNotPaused onlyProvider
_withdrawInTokenFromStrategy	Private	Can Modify State	-
estimateInTokenFromStrategy	External	-	-
balanceOfTokenFromStrategy	External	-	-
setStrategy	External	Can Modify State	onlyOwner
retireStrat	External	Can Modify State	onlyManager
_deactivateStrategy	Private	Can Modify State	-
allocate	Public	Can Modify State	-
_isValidAllocation	Private	-	-
availableToWithdraw	Public	-	-
totalAssets	Public	-	-
totalAssetInVault	Public	-	-
migrateStrategy	External	Can Modify State	onlyManager

MasterVault			
_assessFee	Private	Can Modify State	-
_assessDepositFee	Private	-	-
<Receive Ether>	External	Payable	-
withdrawFee	External	Can Modify State	onlyOwner
setDepositFee	External	Can Modify State	onlyOwner
setWithdrawalFee	External	Can Modify State	onlyOwner
addManager	External	Can Modify State	onlyOwner
removeManager	External	Can Modify State	onlyOwner
changeProvider	External	Can Modify State	onlyOwner
changeFeeReceiver	External	Can Modify State	onlyOwner
changeStrategyAllocation	External	Can Modify State	onlyOwner

WaitingPool			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
addToQueue	External	Can Modify State	onlyMasterVault
tryRemove	External	Can Modify State	onlyMasterVault
<Receive Ether>	External	Payable	-
getPoolBalance	Public	-	-
withdrawUnsettled	External	Can Modify State	-
setCapLimit	External	Can Modify State	onlyMasterVault

ElipsisMediator			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
relyOperator	External	Can Modify State	auth
denyOperator	External	Can Modify State	auth
initialize	Public	Can Modify State	initializer
changeTargetContract	External	Can Modify State	auth
notifyRewardAmount	External	Can Modify State	authOrOperator
setRewardsDuration	External	Can Modify State	authOrOperator

SlidingWindowOracle			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
_authorizeUpgrade	Internal	Can Modify State	onlyOwner
observationIndexOf	Public	-	-
getFirstObservationInWindow	Private	-	-
update	External	Can Modify State	-
computeAmountOut	Private	-	-
consult	External	-	-

UUPSUpgradeable			
Function Name	Visibility	Mutability	Modifiers
__UUPSUpgradeable_init	Internal	Can Modify State	onlyInitializing
__UUPSUpgradeable_init_unchained	Internal	Can Modify State	onlyInitializing

UUPSUpgradeable			
proxiableUUID	External	-	notDelegated
upgradeTo	External	Can Modify State	onlyProxy
upgradeToAndCall	External	Payable	onlyProxy
_authorizeUpgrade	Internal	Can Modify State	-

ERC1967UpgradeUpgradeable			
Function Name	Visibility	Mutability	Modifiers
__ERC1967Upgrade_init	Internal	Can Modify State	onlyInitializing
__ERC1967Upgrade_init_unchained	Internal	Can Modify State	onlyInitializing
_getImplementation	Internal	-	-
_setImplementation	Private	Can Modify State	-
_upgradeTo	Internal	Can Modify State	-
_upgradeToAndCall	Internal	Can Modify State	-
_upgradeToAndCallUUPS	Internal	Can Modify State	-
_getAdmin	Internal	-	-
_setAdmin	Private	Can Modify State	-
_changeAdmin	Internal	Can Modify State	-
_getBeacon	Internal	-	-
_setBeacon	Private	Can Modify State	-
_upgradeBeaconToAndCall	Internal	Can Modify State	-
_functionDelegateCall	Private	Can Modify State	-

PriceOracleTestnet			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
_authorizeUpgrade	Internal	Can Modify State	onlyOwner
peek	Public	-	-

PriceOracleTestnet			

PriceOracle			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
_authorizeUpgrade	Internal	Can Modify State	onlyOwner
peek	Public	-	-

HelioOracle			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
peek	Public	-	-
changePriceToken	External	Can Modify State	-

BusdOracle			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
peek	Public	-	-

BnbOracle			
Function Name	Visibility	Mutability	Modifiers
initialize	External	Can Modify State	initializer
peek	Public	-	-

BaseStrategy			
Function Name	Visibility	Mutability	Modifiers

BaseStrategy			
__BaseStrategy_init	Internal	Can Modify State	initializer
balanceOfWant	Public	-	-
balanceOfPool	Public	-	-
balanceOf	Public	-	-
<Receive Ether>	External	Payable	-
pause	External	Can Modify State	onlyStrategist
unpause	External	Can Modify State	onlyStrategist
setStrategist	External	Can Modify State	onlyOwner
setRewards	External	Can Modify State	onlyOwner

BnbxYieldConverterStrategy			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
deposit	External	Payable	nonReentrant onlyVault
depositAll	External	Can Modify State	nonReentrant onlyStrategist
_deposit	Internal	Can Modify State	whenDepositNotPaused
withdraw	External	Can Modify State	nonReentrant onlyVault
withdrawInToken	External	Can Modify State	nonReentrant onlyVault
estimateInToken	External	-	-
balanceOfToken	External	-	-
panic	External	Can Modify State	nonReentrant onlyStrategist
_withdraw	Internal	Can Modify State	-
batchWithdraw	External	Can Modify State	nonReentrant

BnbxYieldConverterStrategy			
claimNextBatchAndDistribute	External	Can Modify State	nonReentrant
claimNextBatch	Public	Can Modify State	nonReentrant
_claimNextBatch	Private	Can Modify State	-
distributeFund	Public	Can Modify State	nonReentrant
_distributeFund	Private	Can Modify State	-
distributeManual	External	Can Modify State	nonReentrant
harvest	External	Can Modify State	nonReentrant onlyStrategist
_harvestTo	Private	Can Modify State	-
calculateYield	Public	-	-
balanceOfPool	Public	-	-
canDeposit	Public	-	-
assessDepositFee	Public	-	-
changeStakeManager	External	Can Modify State	onlyOwner

CerosYieldConverterStrategy			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
deposit	External	Payable	onlyVault
depositAll	External	Can Modify State	onlyStrategist
_deposit	Internal	Can Modify State	whenDepositNotPaused
withdraw	External	Can Modify State	onlyVault
panic	External	Can Modify State	onlyStrategist
_withdraw	Internal	Can Modify State	-

CerosYieldConverterStrategy			
withdrawInToken	External	Can Modify State	nonReentrant
estimateInToken	External	-	-
balanceOfToken	External	-	-
canDeposit	Public	-	-
assessDepositFee	Public	-	-
harvest	External	Can Modify State	onlyStrategist
_harvestTo	Private	Can Modify State	-
changeBinancePool	External	Can Modify State	onlyOwner
changeCeRouter	External	Can Modify State	onlyOwner

SnBnbYieldConverterStrategy			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
deposit	External	Payable	nonReentrant onlyVault
depositAll	External	Can Modify State	nonReentrant onlyStrategist
_deposit	Internal	Can Modify State	whenDepositNotPaused
withdraw	External	Can Modify State	nonReentrant onlyVault
withdrawInToken	External	Can Modify State	nonReentrant onlyVault
estimateInToken	External	-	-
balanceOfToken	External	-	-
panic	External	Can Modify State	nonReentrant onlyStrategist
_withdraw	Internal	Can Modify State	-
batchWithdraw	External	Can Modify State	nonReentrant

SnBnbYieldConverterStrategy			
claimNextBatchAndDistribute	External	Can Modify State	nonReentrant
claimNextBatch	Public	Can Modify State	nonReentrant
_claimNextBatch	Private	Can Modify State	-
distributeFund	Public	Can Modify State	nonReentrant
_distributeFund	Private	Can Modify State	-
distributeManual	External	Can Modify State	nonReentrant
harvest	External	Can Modify State	nonReentrant onlyStrategist
_harvestTo	Private	Can Modify State	-
calculateYield	Public	-	-
balanceOfPool	Public	-	-
canDeposit	Public	-	-
assessDepositFee	Public	-	-
changeStakeManager	External	Can Modify State	onlyOwner

StkBnbStrategy			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
<Receive Ether>	External	Payable	-
deposit	External	Payable	onlyVault
depositAll	External	Can Modify State	onlyStrategist
_deposit	Internal	Can Modify State	whenDepositNotPaused
withdraw	External	Can Modify State	onlyVault
withdrawInToken	External	Can Modify State	nonReentrant onlyVault

StkBnbStrategy			
balanceOfToken	External	-	-
estimateInToken	External	-	-
panic	External	Can Modify State	onlyStrategist
_withdraw	Internal	Can Modify State	-
claimAndDistribute	External	Can Modify State	-
claimAll	Public	Can Modify State	nonReentrant
claim	External	Can Modify State	nonReentrant
distribute	Public	Can Modify State	nonReentrant
distributeManual	External	Can Modify State	nonReentrant
harvest	External	Can Modify State	nonReentrant onlyStrategist
calculateYield	Public	-	-
balanceOfPool	Public	-	-
canDeposit	Public	-	-
assessDepositFee	Public	-	-
startIndex	External	-	-
endIndex	External	-	-
changeAddressStore	External	Can Modify State	onlyOwner

Clipper			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer

Clipper			
sales	External	-	-
file	External	Can Modify State	auth lock
file	External	Can Modify State	auth lock
min	Internal	-	-
add	Internal	-	-
sub	Internal	-	-
mul	Internal	-	-
wmul	Internal	-	-
rmul	Internal	-	-
rdiv	Internal	-	-
getFeedPrice	Internal	Can Modify State	-
kick	External	Can Modify State	auth lock isStopped
redo	External	Can Modify State	auth lock isStopped
take	External	Can Modify State	auth lock isStopped
_remove	Internal	Can Modify State	-
count	External	-	-
list	External	-	-
getStatus	External	-	-
status	Internal	-	-
upchost	External	Can Modify State	-
yank	External	Can Modify State	auth lock

Dog			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
min	Internal	-	-
add	Internal	-	-
sub	Internal	-	-
mul	Internal	-	-
file	External	Can Modify State	auth
file	External	Can Modify State	auth
file	External	Can Modify State	auth
file	External	Can Modify State	auth
chop	External	-	-
bark	External	Can Modify State	auth
digs	External	Can Modify State	auth
cage	External	Can Modify State	auth
uncage	External	Can Modify State	auth

EmergencyShutdown			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
cage	External	Can Modify State	auth
setMultiSig	External	Can Modify State	onlyOwner

Ownable			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
owner	Public	-	-
renounceOwnership	Public	Can Modify State	onlyOwner
transferOwnership	Public	Can Modify State	onlyOwner
_transferOwnership	Internal	Can Modify State	-

Context			
Function Name	Visibility	Mutability	Modifiers
_msgSender	Internal	-	-
_msgData	Internal	-	-

Flash			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
file	External	Can Modify State	auth
maxFlashLoan	External	-	-
flashFee	External	-	-
flashLoan	External	Can Modify State	nonReentrant
accrue	External	Can Modify State	nonReentrant

Hay			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
transfer	External	Can Modify State	-
transferFrom	Public	Can Modify State	-
mint	External	Can Modify State	auth
burn	External	Can Modify State	-
approve	External	Can Modify State	-
push	External	Can Modify State	-
pull	External	Can Modify State	-
move	External	Can Modify State	-
permit	External	Can Modify State	-
_approve	Internal	Can Modify State	-
increaseAllowance	Public	Can Modify State	-
decreaseAllowance	Public	Can Modify State	-
setSupplyCap	Public	Can Modify State	auth
updateDomainSeparator	External	Can Modify State	auth

HelioRewards			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth

HelioRewards			
initialize	Public	Can Modify State	initializer
initPool	External	Can Modify State	auth
setHelioToken	External	Can Modify State	auth
setRewardsMaxLimit	External	Can Modify State	auth
setOracle	External	Can Modify State	auth
setRate	External	Can Modify State	auth
helioPrice	Public	-	-
rewardsRate	Public	-	-
distributionApy	Public	-	-
pendingRewards	Public	-	-
claimable	Public	-	poolInit
unrealisedRewards	Public	-	poolInit
drop	Public	Can Modify State	-
claim	External	Can Modify State	-
cage	Public	Can Modify State	auth
uncage	Public	Can Modify State	auth

Interaction			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
enableWhitelist	External	Can Modify State	auth
disableWhitelist	External	Can Modify State	auth

Interaction			
setWhitelistOperator	External	Can Modify State	auth
addToWhitelist	External	Can Modify State	operatorOrWard
removeFromWhitelist	External	Can Modify State	operatorOrWard
initialize	Public	Can Modify State	initializer
setCores	Public	Can Modify State	auth
setHayApprove	Public	Can Modify State	auth
setCollateralType	External	Can Modify State	auth
setCollateralDuty	External	Can Modify State	auth
setHelioProvider	External	Can Modify State	auth
removeCollateralType	External	Can Modify State	auth
stringToBytes32	Public	-	-
deposit	External	Can Modify State	whitelisted
borrow	External	Can Modify State	-
dropRewards	Public	Can Modify State	-
payback	External	Can Modify State	-
withdraw	External	Can Modify State	-
drip	Public	Can Modify State	-
poke	Public	Can Modify State	-
setRewards	External	Can Modify State	auth
collateralPrice	Public	-	-
hayPrice	External	-	-
collateralRate	External	-	-
depositTVL	External	-	-

Interaction			
collateralTVL	External	-	-
free	Public	-	-
locked	Public	-	-
borrowed	External	-	-
availableToBorrow	External	-	-
willBorrow	External	-	-
liquidationPriceForDebt	Internal	-	-
currentLiquidationPrice	External	-	-
estimatedLiquidationPrice	External	-	-
estimatedLiquidationPriceHAY	External	-	-
borrowApr	Public	-	-
startAuction	External	Can Modify State	-
buyFromAuction	External	Can Modify State	-
getAuctionStatus	External	-	-
upchostClipper	External	Can Modify State	-
getAllActiveAuctionsForToken	External	-	-
resetAuction	External	Can Modify State	-
totalPegLiquidity	External	-	-
_checkIsLive	Internal	-	-

Jar			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
_min	Internal	-	-

Jar			
lastTimeRewardApplicable	Public	-	-
tokensPerShare	Public	-	-
earned	Public	-	-
replenish	External	Can Modify State	authOrOperator update
setSpread	External	Can Modify State	authOrOperator
setExitDelay	External	Can Modify State	authOrOperator
addOperator	External	Can Modify State	auth
removeOperator	External	Can Modify State	auth
extractDust	External	Can Modify State	auth
cage	External	Can Modify State	auth
uncage	External	Can Modify State	auth
join	External	Can Modify State	update nonReentrant
exit	External	Can Modify State	update nonReentrant
redeemBatch	External	Can Modify State	nonReentrant
_redeemHelper	Private	Can Modify State	-

GemJoin			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
cage	External	Can Modify State	auth
uncage	External	Can Modify State	auth

GemJoin			
join	External	Can Modify State	auth
exit	External	Can Modify State	auth

HayJoin			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
cage	External	Can Modify State	auth
uncage	External	Can Modify State	auth
mul	Internal	-	-
join	External	Can Modify State	auth
exit	External	Can Modify State	auth

Jug			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
_add	Internal	-	-
_diff	Internal	-	-
_rmul	Internal	-	-
init	External	Can Modify State	auth
file	External	Can Modify State	auth

Jug			
file	External	Can Modify State	auth
file	External	Can Modify State	auth
drip	External	Can Modify State	-

Lock			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
file	External	Can Modify State	auth
lockDown	External	Can Modify State	auth
lock	External	Can Modify State	auth
lockExternals	External	Can Modify State	auth
lockCore	External	Can Modify State	auth
unlockAll	External	Can Modify State	auth
unlock	External	Can Modify State	auth
unlockExternals	External	Can Modify State	auth
unlockCore	External	Can Modify State	auth
cage	External	Can Modify State	auth
uncage	External	Can Modify State	auth

Spotter			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
initialize	External	Can Modify State	initializer
mul	Internal	-	-

Spotter			
rdiv	Internal	-	-
file	External	Can Modify State	auth
file	External	Can Modify State	auth
file	External	Can Modify State	auth
poke	External	Can Modify State	-
cage	External	Can Modify State	auth
uncage	External	Can Modify State	auth

Vat			
Function Name	Visibility	Mutability	Modifiers
rely	External	Can Modify State	auth
deny	External	Can Modify State	auth
behalf	External	Can Modify State	auth
regard	External	Can Modify State	auth
hope	External	Can Modify State	-
nope	External	Can Modify State	-
wish	Internal	-	-
initialize	Public	Can Modify State	initializer
_add	Internal	-	-
_sub	Internal	-	-
_mul	Internal	-	-
_add	Internal	-	-
_sub	Internal	-	-

Vat			
_mul	Internal	-	-
init	External	Can Modify State	auth
file	External	Can Modify State	auth
file	External	Can Modify State	auth
cage	External	Can Modify State	auth
uncage	External	Can Modify State	auth
slip	External	Can Modify State	auth
flux	External	Can Modify State	auth
move	External	Can Modify State	auth
either	Internal	-	-
both	Internal	-	-
frob	External	Can Modify State	auth
fork	External	Can Modify State	auth
grab	External	Can Modify State	auth
heal	External	Can Modify State	-
suck	External	Can Modify State	auth
fold	External	Can Modify State	auth

4.3 Vulnerability Summary

[N1] [Medium] Risk of excessive authority

Category: Authority Control Vulnerability Audit

Content

Owner or special administrator accounts can operate the key functions.

```
CeToken  burn
CeToken  mint
CeToken  changeVault

hBNB     burn
hBNB     mint
hBNB     changeMinter

HelioProvider    liquidation
HelioProvider    daoBurn
HelioProvider    daoMint
HelioProvider    changeDao
HelioProvider    changeCeToken
HelioProvider    changeProxy
HelioProvider    changeCollateralToken
HelioProvider    changeOperator

CerosRouter      changeVault
CerosRouter      changeDex
CerosRouter      changePool
CerosRouter      changeProvider

OwnableUpgradeable       renounceOwnership
OwnableUpgradeable       transferOwnership

CeVaultV2        updateStorage

MasterVault      _updateCerosStrategyDebt
MasterVault      depositAllToStrategy
MasterVault      depositToStrategy
MasterVault      withdrawFromStrategy
MasterVault      withdrawAllFromStrategy
MasterVault      setStrategy
MasterVault      retireStrat
MasterVault      migrateStrategy
MasterVault      withdrawFee
MasterVault      setDepositFee
MasterVault      setWithdrawalFee
MasterVault      addManager
MasterVault      removeManager
MasterVault      changeProvider
MasterVault      changeFeeReceiver
MasterVault      changeStrategyAllocation

WaitingPool      addToQueue
WaitingPool      tryRemove
WaitingPool      setCapLimit
```

```
SlidingWindowOracle      _authorizeUpgrade

UUPSUpgradeable  upgradeTo
UUPSUpgradeable  upgradeToAndCall

PriceOracleTestnet       _authorizeUpgrade

PriceOracle      _authorizeUpgrade

BaseStrategy     setStrategist
BaseStrategy     setRewards

BnbxYieldConverterStrategy       changeStakeManager

CerosYieldConverterStrategy      changeBinancePool
CerosYieldConverterStrategy      changeCeRouter

EmergencyShutdown        setMultiSig

Ownable  renounceOwnership
Ownable  transferOwnership

Interaction      addToWhitelist
Interaction      removeFromWhitelist

Jar      replenish
Jar      setSpread
Jar      setExitDelay
```

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

Confirmed; The team will use multiple signatures for the owner to reduce risk, but users should check for themselves that the configuration on the main network is correct.

[N2] [Low] Sandwich attacks can affect slippage scope

Category: Reordering Vulnerability

Content

- contracts/ceros/CerosRouter.sol

```
function deposit()
external
payable
override
nonReentrant
returns (uint256 value)
{
    //...snip code...//
    uint256[] memory outAmounts = _dex.getAmountsOut(amount, path);
    //...snip code...//
    uint256[] memory amounts = _dex.swapExactETHForTokens{
    value: amount
    }(dexABNBcAmount, path, address(this), block.timestamp + 300);
    realAmount = amounts[1];
    //...snip code...//
}

function withdrawWithSlippage(
    address recipient,
    uint256 amount,
    uint256 outAmount
) external override nonReentrant returns (uint256 realAmount) {
    //...snip code...//
    uint256[] memory amounts = _dex.swapExactTokensForETH(
        realAmount,
        outAmount,
        path,
        recipient,
        block.timestamp + 300
    );
    //...snip code...//
}
```

Sandwich attacks, also known as MEV attacks, refer to attackers using the transaction order and execution results on the blockchain to gain additional value. This type of attack is usually carried out by miners or transaction order executors, who can gain additional value by reordering transactions or selectively including or excluding them.

Solution

It is necessary to pay attention to the amount of money passed in during function calls to avoid high slippage.

Status

Acknowledged

[N3] [Critical] Ratio arbitrage attack vulnerability

Category: Unsafe External Call Audit

Content

- contracts/ceros/CeVault.sol
- contracts/ceros/upgrades/CeVaultV2.sol

```
function _deposit(address account, uint256 amount)
private
returns (uint256)
{
    uint256 ratio = _aBNBc.ratio();
    _aBNBc.transferFrom(msg.sender, address(this), amount);
    uint256 toMint = (amount * 1e18) / ratio; //SlowMist//
    _depositors[account] += amount; // aBNBc
    _ceTokenBalances[account] += toMint;
    //  mint ceToken to recipient
    ICertToken(_ceToken).mint(account, toMint);
    emit Deposited(msg.sender, account, toMint);
    return toMint;
}
function _withdraw(
    address owner,
    address recipient,
    uint256 amount
) private returns (uint256) {
    uint256 ratio = _aBNBc.ratio();
    uint256 realAmount = (amount * ratio) / 1e18;//SlowMist//
    require(
        _aBNBc.balanceOf(address(this)) >= realAmount,
        "not such amount in the vault"
    );
    uint256 balance = _ceTokenBalances[owner];
    require(balance >= amount, "insufficient balance");
    _ceTokenBalances[owner] -= amount; // BNB
    // burn ceToken from owner
    ICertToken(_ceToken).burn(owner, amount);
    _depositors[owner] -= realAmount; // aBNBc
    _aBNBc.transfer(recipient, realAmount);
    emit Withdrawn(owner, recipient, realAmount);
```

```
    return realAmount;
}
```

Here we can see that the amount of deposit and withdraw is related to the ratio. We can query the implementation of the ratio from the call chain:

```
_aBNBc:
function ratio() public view returns (uint256) {
    return IBondToken(_bondToken).ratio();
}

_bondToken:
function ratio() public view override returns (uint256) {
    return _ratio;
}
function repairRatio(uint256 newRatio) external onlyOwner {
    _ratio = newRatio;
    emit RatioUpdated(_ratio);
}
function updateRatio(uint256 totalRewards) external onlyOperator {
    uint256 totalShares = totalSharesSupply();
    uint256 denominator = _totalStaked + totalRewards - _totalUnbondedBonds;
    _ratio = multiplyAndDivideFloor(totalShares, 1e18, denominator); // (totalShares
* 1e18) / denominator;
    if (historicalRatios.length == 0) {
        historicalRatios = new uint256[](8);
    }
    if (block.timestamp - _lastUpdate > 1 days - 1 minutes) {
        uint256 _latestOffset = latestOffset;
        historicalRatios[((_latestOffset + 1) % 8)] = _ratio;
        latestOffset = _latestOffset + 1;
        _lastUpdate = block.timestamp;
    }
    emit RatioUpdated(_ratio);
}
```

The value of the ratio can be modified by Owner or through other mechanisms. We may trust the operations of the Owner, but changes in the ratio can cause serious arbitrage attacks that can be implemented without the Owner's permission. The main idea is to use MEV attacks by monitoring the transaction memory pool on the blockchain. When a transaction that increases the ratio is found, one transaction deposits the CeVault contract, and another transaction calls the withdraw function of CeVault. By adjusting the form of the transaction fees, these two

transactions are placed before and after the ratio change transaction, allowing direct get aBNBc in CeVault.

Asset changes like this:

```
Tx1: deposit: 100 aBNBc
Tx2: repairRatio: 1-->1.2
Tx3: withdraw: 120 aBNBc
```

Solution

The design of the Ratio mechanism is not reasonable and needs to be re-evaluated and used.

Status

Fixed; Those functions are used for mock version, which is not existing on live net.

[N4] [High] Missing check return value

Category: Design Logic Audit

Content

Code location:

```
_aBNBc.transferFrom(msg.sender,address(this),amount) (CeVault.sol#70)
_aBNBc.transfer(recipient,availableYields) (CeVault.sol#105)
_aBNBc.transfer(recipient,realAmount) (CeVault.sol#143)

_certToken.transferFrom(owner,address(this),amount) (CerosRouter.sol#125)
_certToken.transfer(recipient,profit) (CerosRouter.sol#165)

IERC20(wBnbToken).approve(dexAddress,type()(uint256).max) (CerosRouter.sol#59)
IERC20(certToken).approve(dexAddress,type()(uint256).max) (CerosRouter.sol#60)
IERC20(certToken).approve(bondToken,type()(uint256).max) (CerosRouter.sol#61)
IERC20(certToken).approve(pool,type()(uint256).max) (CerosRouter.sol#62)
IERC20(certToken).approve(vault,type()(uint256).max) (CerosRouter.sol#63)
_certToken.approve(address(_vault),0) (CerosRouter.sol#250)
_certToken.approve(address(_vault),type()(uint256).max) (CerosRouter.sol#252)
IERC20(_wBnbAddress).approve(address(_dex),0) (CerosRouter.sol#256)
_certToken.approve(address(_dex),0) (CerosRouter.sol#257)
IERC20(_wBnbAddress).approve(address(_dex),type()(uint256).max) (CerosRouter.sol#260)
_certToken.approve(address(_dex),type()(uint256).max) (CerosRouter.sol#261)
_certToken.approve(address(_pool),0) (CerosRouter.sol#266)
_certToken.approve(address(_pool),type()(uint256).max) (CerosRouter.sol#268)

IERC20(_ceToken).approve(daoAddress,type()(uint256).max) (HelioProvider.sol#67)
_ceRouter.withdrawABNBc(recipient,amount) (HelioProvider.sol#155)
_dao.deposit(account,address(_ceToken),amount) (HelioProvider.sol#174)
```

```
_dao.withdraw(account,address(_ceToken),amount) (HelioProvider.sol#178)
IERC20(_ceToken).approve(address(_dao),0) (HelioProvider.sol#194)
IERC20(_ceToken).approve(address(_dao),type()(uint256).max) (HelioProvider.sol#196)
IERC20(_ceToken).approve(address(_dao),0) (HelioProvider.sol#200)
IERC20(_ceToken).approve(address(_dao),type()(uint256).max) (HelioProvider.sol#202)

IERC20Upgradeable(_rewardsToken).approve(address(target),reward)
(mediator/ElipsisMediator.sol#59)

_bnbxToken.approve(destination,type()(uint256).max)
(strategy/BnbxYieldConverterStrategy.sol#58)
_bnbxToken.approve(address(_stakeManager),0)
(strategy/BnbxYieldConverterStrategy.sol#313)
_bnbxToken.approve(address(_stakeManager),type()(uint256).max)
(strategy/BnbxYieldConverterStrategy.sol#315)

_certToken.approve(binancePool,type()(uint256).max)
(strategy/CerosYieldConverterStrategy.sol#40)
_certToken.approve(address(_binancePool),0)
(strategy/CerosYieldConverterStrategy.sol#146)
_certToken.approve(address(_binancePool),type()(uint256).max)
(strategy/CerosYieldConverterStrategy.sol#148)

_snBnbToken.approve(destination,type()(uint256).max)
(strategy/SnBnbYieldConverterStrategy.sol#57)
_snBnbToken.approve(address(_stakeManager),0)
(strategy/SnBnbYieldConverterStrategy.sol#309)
_snBnbToken.approve(address(_stakeManager),type()(uint256).max)
(strategy/SnBnbYieldConverterStrategy.sol#311)

hay.transferFrom(address(receiver), address(this), total)(contracts/flash.sol#110)

hay.transfer(keeper,hayBal) (libraries/AuctionProxy.sol#39)
hay.transfer(keeper,hayBal) (libraries/AuctionProxy.sol#62)
hay.transferFrom(msg.sender,address(this),hayMaxAmount)
(libraries/AuctionProxy.sol#83)
hay.transfer(receiverAddress,hayBal) (libraries/AuctionProxy.sol#99)

hay.approve(hayJoin_,type()(uint256).max) (Interaction.sol#103)
hay.approve(address(hayJoin),0) (Interaction.sol#109)
hay.approve(hayJoin_,type()(uint256).max) (Interaction.sol#118)
hay.approve(address(hayJoin),type()(uint256).max) (Interaction.sol#122)

IERC20Upgradeable(hay).transferFrom(msg.sender,address(this),wad) (vow.sol#99)
```

Not verifying the return value may lead to logical errors.

Solution

Check return value.

Status

Fixed

[N5] [Medium] `transfer()` and `send()` risks

Category: Denial of Service Vulnerability

Content

Any smart contract that uses transfer() or send() is taking a hard dependency on gas costs by forwarding a fixed amount of gas: 2300.

Gas costs are not constant. Smart contracts should be robust to this fact.

Code location:

```
contracts/masterVault/MasterVault.sol#L138
contracts/masterVault/MasterVault.sol#L142
contracts/masterVault/MasterVault.sol#L437
contracts/strategy/CerosYieldConverterStrategy.sol#L82
```

Reference:

https://consensys.net/diligence/blog/2019/09/stop-using-soliditys-transfer-now/

Solution

Stop using transfer() and send() in your code and switch to using call() instead. This carries a risk regarding reentrancy. Be sure to use one of the robust methods available for preventing reentrancy vulnerabilities.

Status

Fixed

[N6] [Suggestion] Missing zero address validation

Category: Design Logic Audit

Content

Code locations:

```
CeToken.changeVault(address).vault (CeToken.sol#47)
CeVault.changeRouter(address).router (CeVault.sol#194)
```

```
CerosRouter.changeProvider(address).provider (CerosRouter.sol#271)
hBNB.changeMinter(address).minter (hBNB.sol#42)

HelioProvider.initialize(address,address,address,address,address,address).certToken
(HelioProvider.sol#51)
HelioProvider.initialize(address,address,address,address,address,address).ceToken
(HelioProvider.sol#52)
HelioProvider.changeCeToken(address).ceToken (HelioProvider.sol#199)
HelioProvider.changeProxy(address).auctionProxy (HelioProvider.sol#205)
HelioProvider.changeOperator(address).operator (HelioProvider.sol#213)

MasterVault.initialize(uint256,uint256,uint8,address,address).ceToken
(masterVault/MasterVault.sol#86)
MasterVault.withdrawETH(address,uint256).account (masterVault/MasterVault.sol#126)

PriceOracle.initialize(address,IMovingWindowOracle,bool,address,address)._tokenIn
(oracle/PriceOracle.sol#22)
PriceOracle.initialize(address,IMovingWindowOracle,bool,address,address)._wbnb
(oracle/PriceOracle.sol#25)
PriceOracle.initialize(address,IMovingWindowOracle,bool,address,address)._usd
(oracle/PriceOracle.sol#26)
PriceOracleTestnet.initialize(address,IMovingWindowOracle,bool)._tokenIn
(oracle/PriceOracleTestnet.sol#31)

SlidingWindowOracle.initialize(address,uint256,uint8).factory_
(oracle/SlidingWindowOracle.sol#46)

BnbxYieldConverterStrategy.distributeManual(address).recipient
(strategy/BnbxYieldConverterStrategy.sol#253)

SnBnbYieldConverterStrategy.distributeManual(address).recipient
(strategy/SnBnbYieldConverterStrategy.sol#253)

StkBnbStrategy.distributeManual(address).recipient (strategy/StkBnbStrategy.sol#280)

Interaction.setWhitelistOperator(address).usr (Interaction.sol#59)
Interaction.initialize(address,address,address,address,address,address,address).dog_
(Interaction.sol#86)

Jar.initialize(string,string,address,uint256,uint256,uint256)._hayToken (jar.sol#88)

Clipper.file(bytes32,address).data (clip.sol#166)

Dog.file(bytes32,address).data (dog.sol#133)

EmergencyShutdown.constructor(address,address)._vat (es.sol#19)
EmergencyShutdown.constructor(address,address)._multisig (es.sol#19)
EmergencyShutdown.setMultiSig(address)._multisig (es.sol#28)
```

```
Vow.initialize(address,address,address)._hayJoin (vow.sol#59)
Vow.initialize(address,address,address).multisig_ (vow.sol#59)
Vow.file(bytes32,address).data (vow.sol#78)
```

Solution

Check that the address is not zero.

Status

Fixed

[N7] [Low] Performs a multiplication on the result of a division

Category: Arithmetic Accuracy Deviation Vulnerability

Content

```
- price = oneTokenOut / amountOut * 10 ** 18 (oracle/PriceOracleTestnet.sol#55)

- d /= pow2 (oracle/libraries/FullMath.sol#23)
- r *= 2 - d * r (oracle/libraries/FullMath.sol#27)

- d /= pow2 (oracle/libraries/FullMath.sol#23)
- r *= 2 - d * r (oracle/libraries/FullMath.sol#28)

- d /= pow2 (oracle/libraries/FullMath.sol#23)
- r *= 2 - d * r (oracle/libraries/FullMath.sol#29)

- d /= pow2 (oracle/libraries/FullMath.sol#23)
- r *= 2 - d * r (oracle/libraries/FullMath.sol#30)

- d /= pow2 (oracle/libraries/FullMath.sol#23)
- r *= 2 - d * r (oracle/libraries/FullMath.sol#31)

- d /= pow2 (oracle/libraries/FullMath.sol#23)
- r *= 2 - d * r (oracle/libraries/FullMath.sol#32)

- d /= pow2 (oracle/libraries/FullMath.sol#23)
- r *= 2 - d * r (oracle/libraries/FullMath.sol#33)

- d /= pow2 (oracle/libraries/FullMath.sol#23)
- r *= 2 - d * r (oracle/libraries/FullMath.sol#34)

- l /= pow2 (oracle/libraries/FullMath.sol#24)
- l * r (oracle/libraries/FullMath.sol#35)

- poolTokens = (poolTokensToBurn * 1e11) / (1e11 -
```

```
stakePool.config().fee.withdraw) (strategy/StkBnbStrategy.sol#188)
    - poolTokensFee = (poolTokens * stakePool.config().fee.withdraw) / 1e11
(strategy/StkBnbStrategy.sol#198)

    - x = xxRound_rpow_asm_0 / b (hMath.sol#19)
    - zx_rpow_asm_0 = z * x (hMath.sol#21)

    - d /= pow2 (oracle/libraries/FullMath.sol#23)
    - r *= 2 - d * r (oracle/libraries/FullMath.sol#27)

    - d /= pow2 (oracle/libraries/FullMath.sol#23)
    - r *= 2 - d * r (oracle/libraries/FullMath.sol#28)

    - d /= pow2 (oracle/libraries/FullMath.sol#23)
    - r *= 2 - d * r (oracle/libraries/FullMath.sol#29)

    - d /= pow2 (oracle/libraries/FullMath.sol#23)
    - r *= 2 - d * r (oracle/libraries/FullMath.sol#30)

    - d /= pow2 (oracle/libraries/FullMath.sol#23)
    - r *= 2 - d * r (oracle/libraries/FullMath.sol#31)

    - d /= pow2 (oracle/libraries/FullMath.sol#23)
    - r *= 2 - d * r (oracle/libraries/FullMath.sol#32)

    - d /= pow2 (oracle/libraries/FullMath.sol#23)
    - r *= 2 - d * r (oracle/libraries/FullMath.sol#33)

    - d /= pow2 (oracle/libraries/FullMath.sol#23)
    - r *= 2 - d * r (oracle/libraries/FullMath.sol#34)

    - l /= pow2 (oracle/libraries/FullMath.sol#24)
    - l * r (oracle/libraries/FullMath.sol#35)

    - x = xxRound_rpow_asm_0 / b (abaci.sol#159)
    - zx_rpow_asm_0 = z * x (abaci.sol#161)

    - x = xxRound_rpow_asm_0 / b (abaci.sol#249)
    - zx_rpow_asm_0 = z * x (abaci.sol#251)

    - rate = wad / timeline (jar.sol#141)
    - leftover = remaining * rate (jar.sol#144)

    - x = xxRound_rpow_asm_0 / b (hMath.sol#19)
    - zx_rpow_asm_0 = z * x (hMath.sol#21)
```

Solidity's integer division truncates. Thus, performing division before multiplication can lead to precision loss.

Solution

Consider ordering multiplication before division.

Status

Ignored

[N8] [Low] Anyone can call initialize on the logic contract

Category: Race Conditions Vulnerability

Content

```
contracts/ceros/upgrades/CeVaultV2.sol
  33,5:     function initialize(

contracts/ceros/upgrades/HelioProviderV2.sol
  43,5:     function initialize(

contracts/ceros/CerosRouter.sol
  41,5:     function initialize(

contracts/ceros/CeToken.sol
  31,5:     function initialize(string calldata _name, string calldata _symbol)

contracts/ceros/CeVault.sol
  33,5:     function initialize(

contracts/ceros/hBNB.sol
  29,5:     function initialize() external initializer {

contracts/ceros/HelioProvider.sol
  49,5:     function initialize(

contracts/masterVault/MasterVault.sol
  82,5:     function initialize(

contracts/masterVault/WaitingPool.sol
  28,5:     function initialize(address _masterVault, uint256 _capLimit) external
initializer {

contracts/mediator/ElipsisMediator.sol
  43,5:     function initialize(address targetContract) public initializer {

contracts/oracle/interfaces/IUniswapV2Pair.sol
  96,3:   function initialize(address, address) external;
```

```
contracts/oracle/BnbOracle.sol
  11,5:     function initialize(address aggregatorAddress) external initializer {

contracts/oracle/BusdOracle.sol
  11,5:     function initialize(address aggregatorAddress) external initializer {

contracts/oracle/HelioOracle.sol
  17,5:     function initialize(uint256 initialPrice) public initializer {

contracts/oracle/PriceOracle.sol
  21,3:   function initialize(

contracts/oracle/PriceOracleTestnet.sol
  30,3:   function initialize(

contracts/oracle/SlidingWindowOracle.sol
  45,3:   function initialize(

contracts/strategy/BnbxYieldConverterStrategy.sol
  46,5:     function initialize(

contracts/strategy/CerosYieldConverterStrategy.sol
  27,5:     function initialize(

contracts/strategy/SnBnbYieldConverterStrategy.sol
  45,5:     function initialize(

contracts/strategy/StkBnbStrategy.sol
  58,5:     function initialize(

contracts/abaci.sol
  52,5:     function initialize() external initializer {
  123,5:     function initialize() external initializer {
  214,5:     function initialize() external initializer {

contracts/clip.sol
  127,5:     function initialize(address vat_, address spotter_, address dog_,
bytes32 ilk_) external initializer {

contracts/dog.sol
  103,5:     function initialize(address vat_) external initializer {

contracts/flash.sol
  58,5:     function initialize(address _vat, address _hay, address _hayJoin, address
_vow) external initializer {

contracts/hay.sol
  59,5:     function initialize(uint256 chainId_, string memory symbol_, uint256
supplyCap_) external initializer {
```

```
contracts/HelioRewards.sol
  65,5:     function initialize(address vat_, uint256 poolLimit_ ) public initializer
{

contracts/HelioToken.sol
  28,5:     function initialize(uint256 rewardsSupply_, address rewards_) public
initializer {

contracts/Interaction.sol
  80,5:     function initialize(

contracts/jar.sol
  88,5:     function initialize(string memory _name, string memory _symbol, address
_hayToken, uint _spread, uint _exitDelay, uint _flashLoanDelay) external initializer
{

contracts/join.sol
  89,5:     function initialize(address vat_, bytes32 ilk_, address gem_) external
initializer {
  149,5:     function initialize(address vat_, address hay_) external initializer {

contracts/jug.sol
  54,5:     function initialize(address vat_) external initializer {

contracts/lock.sol
  64,5:     function initialize() external initializer {

contracts/spot.sol
  56,5:     function initialize(address vat_) external initializer {

contracts/vat.sol
  74,5:     function initialize() public initializer {

contracts/vow.sol
  59,5:     function initialize(address vat_, address _hayJoin, address multisig_)
external initializer {
```

Anyone can call initialize on the logic contract.

Solution

Add a constructor to ensure initialize cannot be called on the logic contract.

Status

Acknowledged

[N9] [High] vaultToken burned may exceed the actual number needed

Category: Design Logic Audit

Content

- contracts/masterVault/MasterVault.sol

```
function withdrawETH(address account, uint256 amount)
external
override
nonReentrant
whenNotPaused
onlyProvider
returns (uint256 shares) {
    address src = msg.sender;
    ICertToken(vaultToken).burn(src, amount);
    uint256 ethBalance = totalAssetInVault();
    shares = _assessFee(amount, withdrawalFee);
    if(ethBalance < shares) {
        payable(account).transfer(ethBalance);
        uint256 withdrawn = withdrawFromActiveStrategies(account, shares -
ethBalance);
        shares = ethBalance + withdrawn;
    } else {
        payable(account).transfer(shares);
    }
    emit Withdraw(src, src, src, amount, shares);
    return amount;
}
```

When Strategy balance is not enough, the actual `withdrawn` amount return by `withdrawFromActiveStrategies` will lower than `shares - ethBalance` passed, it means the provider burn `amount` but do not get enough native token.

Solution

Burn less token if Strategy balance is not enough.

Status

Fixed

[N10] [High] `_rewardsToken` can be arbitrary contract

Category: Design Logic Audit

Content

- contracts/mediator/ElipsisMediator.sol

```
function notifyRewardAmount(address _rewardsToken, uint256 reward) external
authOrOperator {
    require(live == 1, "mediator/not-live");
    IERC20Upgradeable(_rewardsToken).safeTransferFrom(msg.sender, address(this),
reward);
    IERC20Upgradeable(_rewardsToken).approve(address(target), reward);
    target.notifyRewardAmount(_rewardsToken, reward);
}
```

`_rewardsToken` can be arbitrary contract, malicious user can attack `target` contract by pass an malicious `_rewardsToken` address.

Solution

Check `_rewardsToken`

Status

Fixed

[N11] [Medium] Missing check BnbOracle status

Category: Unsafe External Call Audit

Content

- contracts/oracle/BnbOracle.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);
}
```

In order to get a correct price, we need to check key values returned by priceFeed.

Solution

Check `roundID/startedAt/timeStamp/answeredInRound`

Status

Fixed

[N12] [Suggestion] Upgrading contracts may introduce new risks

Category: Authority Control Vulnerability Audit

Content

- contracts/oracle/HelioOracle.sol
- contracts/oracle/PriceOracle.sol
- contracts/oracle/SlidingWindowOracle.sol

The Proxy can upgrade the contract by calling `upgradeTo/upgradeToAndCall`, and upgrading the contract may introduce new risks.

Solution

Rigorous auditing of contract upgrades.

Status

Acknowledged; The team have multi-sig signature policy to upgrade contracts on live net.

[N13] [High] HelioOracle owner is never initialized

Category: Design Logic Audit

Content

- contracts/oracle/HelioOracle.sol

```
function changePriceToken(uint256 price_) external {
    require(msg.sender == _owner, "HelioOracle/forbidden");
    price = price_;
```

```
    emit PriceChanged(price);
}
```

`_owner` is never initialized, changePriceToken call will fail in any condition.

Solution

Initiallize `_owner`

Status

Fixed

[N14] [Low] HelioOracle price oracle is not rigorous

Category: Design Logic Audit

Content

- contracts/oracle/HelioOracle.sol

```
function changePriceToken(uint256 price_) external {
    require(msg.sender == _owner, "HelioOracle/forbidden");
    price = price_;
    emit PriceChanged(price);
}
```

This oracle price is too simple, there is not parameters for determining the validity of prices, such as timestamp.

Solution

Use a more rigorous price oracle.

Status

Acknowledged

[N15] [Medium] Oracle price should not return 0

Category: Unsafe External Call Audit

Content

- contracts/HelioRewards.sol

```
function helioPrice() public view returns(uint256) {
    // 1 HAY is helioPrice() helios
```

```
    (bytes32 price, bool has) = oracle.peek();
    if (has) {
        return uint256(price);
    } else {
        return 0;
    }
}
```

- contracts/Interaction.sol

```
function collateralPrice(address token) public view returns (uint256) {
    CollateralType memory collateralType = collaterals[token];
    _checkIsLive(collateralType.live);

    (PipLike pip,) = spotter.ilks(collateralType.ilk);
    (bytes32 price, bool has) = pip.peek();
    if (has) {
        return uint256(price);
    } else {
        return 0;
    }
}
```

Price oracle should break the operation when peek an error, instead of return 0.

Solution

require(has, "invalid-price");

Status

Fixed

[N16] [Medium] ERC777 reentrancy risks

Category: Reentrancy Vulnerability

Content

- contracts/strategy/StkBnbStrategy.sol

```
function _withdraw(address recipient, uint256 amount) internal returns (uint256) {
    //...
    stkBNB.send(address(stakePool), poolTokens, "");

    // save it so that we can later dispatch the amount to the recipient on claim
    withdrawReqs[_endIndex++] = WithdrawRequest(recipient, value);
```

```
    // keep track of _netDeposits in StakePool
    _bnbDepositsInStakePool -= value;

    return value + ethBalance;
}
```

stkBNB is a ERC777 token , ERC777 tokens are vulnerable to reentrancy attacks due to a design flaw.

Solution

Add `nonReentrant` modifier.

Status

Fixed

[N17] [Suggestion] Missing events access control

Category: Malicious Event Log Audit

Content

```
MasterVault._updateCerosStrategyDebt()
MasterVault.withdrawFee()
BnbxYieldConverterStrategy._deposit()
BnbxYieldConverterStrategy._withdraw()
BnbxYieldConverterStrategy._distributeFund()
BnbxYieldConverterStrategy._harvestTo()
CerosYieldConverterStrategy._deposit()
CerosYieldConverterStrategy._withdraw()
CerosYieldConverterStrategy._harvestTo()
SnBnbYieldConverterStrategy._deposit()
SnBnbYieldConverterStrategy.withdrawInToken()
SnBnbYieldConverterStrategy._withdraw()
SnBnbYieldConverterStrategy._distributeFund()
SnBnbYieldConverterStrategy._harvestTo()
StkBnbStrategy._deposit()
StkBnbStrategy.withdrawInToken()
StkBnbStrategy._withdraw()
StkBnbStrategy.harvest()
```

Solution

Emit an event for critical parameter changes.

Status

Fixed

[N18] [Suggestion] Reentry prevention best practices

Category: Reentrancy Vulnerability

Content

- contracts/HelioRewards.sol

```
function claim(uint256 amount) external {
    //...
}
```

- contracts/Interaction.sol

```
function deposit(
    address participant,
    address token,
    uint256 dink
) external whitelisted(participant) returns (uint256) {
}
function borrow(address token, uint256 hayAmount) external returns (uint256) {
    //...
}

function payback(address token, uint256 hayAmount) external returns (int256) {
    //...
}
```

Not apply check-effects-interactions pattern when making external calls in these functions.

Solution

It's recommend to add `nonReentrant` modifier in key places.

Status

Fixed

[N19] [Low] Missing check return value

Category: Design Logic Audit

Content

Code location:

```
jug.drip(collateralType.ilk) (Interaction.sol#145)
jug.drip(collateralType.ilk) (Interaction.sol#309)

_deactivateStrategy(strategy)(contracts/masterVault/MasterVault.sol#315)
_depositToStrategy(strategies[i], depositAmount)
(contracts/masterVault/MasterVault.sol#342)
```

Not verifying the return value may lead to logical errors.

Solution

Check return value.

Status

Acknowledged

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002306010001	SlowMist Security Team	2023.05.12 - 2023.06.01	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 critical risk, 4 high risk, 5 medium risk, 5 low risk, 4 suggestion vulnerabilities. And 1 medium risk vulnerabilities were confirmed and being fixed; 1 low risk vulnerabilities were ignored;

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