智能合约安全审计报告

零时科技
NONEAGE

Rollup.Finance 智能合约安全审计报告

审计团队：零时科技安全团队

审计时间：2023-04-24

目录

Rollup.Finance 智能合约安全审计报告

1.概述

零时科技安全团队于 2023 年 04 月 15 日，接到 **Rollup.Finance 项目**的安全审计需求，团队于 2023 年 04 月 24 日对 **Rollup.Finance 智能合约**审计完成，审计过程中零时科技安全审计专家与 Rollup.Finance 项目接口人员进行沟通，并保持信息对称，在操作风险可控的情况下进行安全审计工作，规避在测试过程中对项目产生和运营造成风险。

经过与 Rollup.Finance 项目方沟通反馈，确认审计过程中发现的漏洞及风险均已修复或在可承受范围内，本次 Rollup.Finance 智能合约安全审计结果：通过安全审计。

合约报告 Hash：
F3AB7508546AFBEEC1575177681DDF2B46D16D2A39B5D4A9DAE30A717666C44C

2.项目背景

2.1 项目简介

项目名称：Rollup.Finance

项目官网：https://rollup.finance

合约类型：永续合约

代码语言：Solidity

合约文件：YieldToken.sol,USDR.sol,LP.sol,WETH.sol,BaseToken.sol,MintableBaseToken.sol,FaucetToken.sol,Multicall.sol,TokenManager.sol,Timelock.sol,Governable.sol,Reader.sol,VaultReader.sol,BalanceUpdater.sol,BatchSender.sol,RewardReader.sol,OrderBookReader.sol,DexV3Aggregator.sol,FastPriceEvents.sol,CustomV3Aggregator.sol,ConstantV3Aggregator.sol,FastPriceFeed.sol,VaultWrapper.sol,PositionRouter.sol,VaultPriceFeed.sol,PositionManager.sol,ShortsTracker.sol,OrderBook.sol,Vault.sol,Router.sol,BasePositionManager.sol,RewardTracker.sol,RewardRouterV1.sol,RewardDistributor.sol,ReferralStorage.sol,ReferralReader.sol

2.2 审计范围

Rollup.Finance 官方提供合约文件及文件对应 SHA256：

YieldToken.sol	934EB8FEE29BD2718D1BB3AEA0A1C2385216AF2CA29806842210617204４C0B31
USDR.sol	8AF1706CC15BBE5A91CC5E79014AFA265F32DA97C973F96558E9E65AA96A7BBF
LP.sol	EBEA9C95469DE9A86F58A1BDB0FC0D1CB9CC248520AC387036D86304B19BB551
WETH.sol	05F87C74ECFE266BC3D70534ADA651DDBA2BE1CEE82031A36B11F1A51E0D3755
BaseToken.sol	8B8BD15090EB891BB95AC0DCCA9F95EF2465E010608202D5A1D7B99D2AF24AA8
MintableBaseToken.sol	5DC1C857DEBA7C4250011273C58E9967EBBA2512E17F0E5FEB1D3A85E0DB240D
FaucetToken.sol	DC0CC81B20EC0FDC012248D39C7041A27FD327D463FA0072B7411E4FF7B0DA4C
Multicall.sol	BE0A4CBE03A9C47D464E28405A772BD702EFF80E6ED97A0A938C733DAADDBB57
TokenManager.sol	B3F53C9F973AC600D5A4CD877230FF69179F094DDA1A32C12202612F6620D2E1
Governable.sol	A002AFCEF81A5743C542E2BBF1E750A311BF87F32F8DD1A8E0CA8F3E346012AB
Reader.sol	76AABF02BD8C349CC13F7F3D0958E2F0EB7289DB451C53F68F57CA6E62F036E6
VaultReader.sol	ADDA6E32BAE2CB44ACD5A8AEE1CCF68C405FFA3C7373C90B7296481792871FC8

BalanceUpdater.sol	759139F23E3F3424076E5AF2FFF9B0581906E97297AB72443AB82332617D1965
BatchSender.sol	3E19A8036C2496BFC1AA4A0F939A43B6FF0B862B0AFB3D73509F89C996165C6F
RewardReader.sol	C87F4E8CFB4CEEE5CA0EE8BBAA46A198AAFA0D0841AC3CB9A9054A1F05A16011
OrderBookReader.sol	CC39BE8F62078DB529F4D7D658FDCED2994B7146D83F8FE965640E565F6EBF2D
DexV3Aggregator.sol	AEFD194DD07340BA85AB1F776D90B1B39BF951F39946582E5AC4506CE894A400
FastPriceEvents.sol	6BD8D2795D3C9191CB4ACFBD0CA15B612EA793562CB1286A195C47CD08F1BE25
CustomV3Aggregator.sol	8EB250A2AC820D75E7CAE934954FE7D3C18D8C2DD8E1D093EB56D7C674257A4E
ConstantV3Aggregator.sol	EC610AA475FFCFFD21BF657819DD3031D41F98F1201B7B63E20FDF1093A40134
FastPriceFeed.sol	108F50885C9893BB2F7450A80F2A52285B9397B57AE1BA645EF80356A1B1366C
LpManager.sol	8E2E3A74FE00D34DEE87A9877D6F676AA3777D7B7F785E818E6230F2B5C4A18F
VaultWrapper.sol	3616E780E153CD79BA2623A3D2BD09834D6F790A8B071384EFDB9724997C05B3
PositionRouter.sol	ED4A4EC4164799BA8F637080B70A1C4B47B07E6683F8C34637110A100E06B994
VaultPriceFeed.sol	A9FCFDACBD34023C562B0852DDCA7FE103926145F8C3A2028760F8173BF4958F

PositionManager.sol	C4C97BCCFF56693DB783215C49D7BB06A0F289E9E6AEBFDDAD2C10F1CAAB9073
ShortsTracker.sol	8D1985A308CE98951F4BD7E11DFB119EE38DEF86B2C2BC331CB0F518A1B23FA6
OrderBook.sol	05A2B83DEAF0C37F66416B3EC28F0D84ACD551808D10FC19F007E5CA80815937
Vault.sol	01601536D9D03F7E5E7142747D3A73EEABB319FDEEC9597CAC336DC7733C04BC
Router.sol	C0165FAD09F57065B35426D548212B057534F08CF68F54062973B28D264FD15B
BasePositionManager.sol	9700AFDC5F423F0B5CF84788F7A71B3F60D954EC8FAA7F66002746B7A50B0E67
RewardTracker.sol	27382DB51E3C7232C9EEE3D688B818B75F9F1405DBB992664A80DC379AE0E336
RewardRouterV1.sol	179F52121E97ED4C5B0200F69CA8CB06E2DE1E1FC97EE2853D0A40D0C4D99C10
RewardDistributor.sol	4CF95FE5BABD593BB73AC74EF5B325F2C3BF3A7C89F639F03AEC9137BED8EE4A

3.合约架构分析

3.1 目录结构

```
├──── access
|    └──── Governable.sol
├──── core
|    ├──── BasePositionManager.sol
|    ├──── LpManager.sol
|    ├──── OrderBook.sol
|    ├──── PositionManager.sol
|    ├──── PositionRouter.sol
|    ├──── Router.sol
|    ├──── ShortsTracker.sol
|    ├──── VaultPriceFeed.sol
|    ├──── Vault.sol
|    └──── VaultWrapper.sol
├──── Multicall.sol
├──── oracle
|    ├──── ConstantV3Aggregator.sol
|    ├──── CustomV3Aggregator.sol
|    ├──── DexV3Aggregator.sol
|    ├──── FastPriceEvents.sol
|    └──── FastPriceFeed.sol
├──── peripherals
|    ├──── BalanceUpdater.sol
|    ├──── BatchSender.sol
```

```
│   ├── OrderBookReader.sol
│   ├── Reader.sol
│   ├── RewardReader.sol
│   └── VaultReader.sol
├── referrals
│   ├── ReferralReader.sol
│   └── ReferralStorage.sol
├── staking
│   ├── RewardDistributor.sol
│   ├── RewardRouterV1.sol
│   └── RewardTracker.sol
├── timelock
│   ├── Timelock.sol
│   └── TokenManager.sol
└── tokens
    ├── BaseToken.sol
    ├── FaucetToken.sol
    ├── LP.sol
    ├── MintableBaseToken.sol
    ├── USDR.sol
    ├── WETH.sol
    └── YieldToken.sol
```

3.2 合约详情

Multicall Contract

方法名称	方法传参	方法属性
aggregate	Call[] memory calls	public
getEthBalance	address addr	public
getBlockHash	uint256 blockNumber	public
getLastBlockHash	none	public
getCurrentBlockTimestamp	none	public
getCurrentBlockDifficulty	none	public
getCurrentBlockGasLimit	none	public
getCurrentBlockCoinbase	none	public

YieldToken Contract

方法名称	方法传参	方法属性
setGov	address _gov	onlyGov
setInfo	string _name string _symbol	onlyGov
setYieldTrackers	address[] memory _yieldTrackers	onlyGov
addAdmin	address _account	onlyGov
removeAdmin	address _account	onlyGov
withdrawToken	address _token address _account uint256 _amount	onlyGov
setInWhitelistMode	bool _inWhitelistMode	onlyGov
setWhitelistedHandler	address _handler bool _isWhitelisted	onlyGov
addNonStakingAccount	address _account	onlyAdmin
removeNonStakingAccount	address _account	onlyAdmin
recoverClaim	address _account address _receiver	onlyAdmin

claim	address _receiver	external
totalStaked	none	external
balanceOf	address _account	external
stakedBalance	address _account	external
transfer	address _recipient uint256 _amount	external
allowance	address _owner address _spender	external
approve	address _spender uint256 _amount	external
transferFrom	address _sender address _recipient uint256 _amount	external
_mint	address _account uint256 _amount	internal
_burn	address _account uint256 _amount	internal
_transfer	address _sender address _recipient uint256 _amount	private
_approve	address _owner address _spender uint256 _amount	private
_updateRewards	address _account	private

USDR Contract

方法名称	方法传参	方法属性
addVault	address _vault	onlyGov
removeVault	address _vault	onlyGov
mint	address _account uint256 _amount	onlyVault
Burn	address _account uint256 _amount	onlyVault

LP Contract

方法名称	方法传参	方法属性
id	none	external

WETH Contract

方法名称	方法传参	方法属性
deposit	none	public
withdraw	uint256 amount	public
name	none	public
symbol	none	public
decimals	none	public
totalSupply	none	public
balanceOf	address account	public
transfer	address recipient uint256 amount	public
allowance	address owner address spender	public

approve	address spender uint256 amount	public
transferFrom	address sender address recipient uint256 amount	public
increaseAllowance	address spender uint256 addedValue	public
decreaseAllowance	address spender uint256 subtractedValue	public
_transfer	address sender address recipient uint256 amount	internal
_mint	address account uint256 amount	internal
_burn	address account uint256 amount	internal
_approve	address owner address spender uint256 amount	internal
_beforeTokenTransfer	address from address to uint256 amount	internal
_msgSender	none	internal

BaseToken Contract

方法名称	方法传参	方法属性
setGov	address _gov	onlyGov
setInfo	string _name string _symbol	onlyGov
setYieldTrackers	address[] memory _yieldTrackers	onlyGov
addAdmin	address _account	onlyGov
removeAdmin	address _account	onlyGov

withdrawToken	address _token address _accoun uint256 _amount	onlyGov
setInPrivateTransferMode	bool _inPrivateTransferMode	onlyGov
setHandler	address _handler bool _isActive	onlyGov
addNonStakingAccount	address _account	onlyAdmin
removeNonStakingAccount	address _account	onlyAdmin
recoverClaim	address _account address _receiver	onlyAdmin
claim	address _receiver	external
totalStaked	none	external
balanceOf	address _account	external
stakedBalance	address _account	external
transfer	address _recipient uint256 _amount	external
allowance	address _owner address _spender	external
approve	address _spender uint256 _amount	external
transferFrom	address _sender address _recipient uint256 _amount	external
_mint	address _account uint256 _amount	internal
_burn	address _account uint256 _amount	internal
_transfer	address _sender address _recipient uint256 _amount	private
_approve	address _owner address _spender uint256 _amount	private
_updateRewards	address _account	private

MintableBaseToken Contract

方法名称	方法传参	方法属性
setMinter	address _minter bool _isActive	onlyGov
mint	address _account uint256 _amount	onlyMinter
burn	address _account uint256 _amount	onlyMinter

FaucetToken Contract

方法名称	方法传参	方法属性
mint	address account uint256 amount	public
enableFaucet	none	public
disableFaucet	none	public
setDropletAmount	uint256 dropletAmount	public
claimDroplet	none	public
name	none	public
symbol	none	public
decimals	none	public
totalSupply	none	public
balanceOf	address account	public
transfer	address recipient uint256 amount	public
allowance	address owner address spender	public
approve	address spender uint256 amount	public
transferFrom	address sender address recipient uint256 amount	public
increaseAllowance	address spender uint256 addedValue	public

decreaseAllowance	address spender uint256 subtractedValue	public
_transfer	address sender address recipient uint256 amount	internal
_mint	address account uint256 amount	internal
_burn	address account uint256 amount	internal
_approve	address owner address spender uint256 amount	internal
_beforeTokenTransfer	address from address to uint256 amount	internal
_msgSender	none	internal

TokenManager Contract

方法名称	方法传参	方法属性
initialize	address[] memory _signers	onlyAdmin
signersLength	none	public
signalApprove	address _token address _spender uint256 _amount	onlyAdmin
signApprove	address _token address _spender uint256 _amount uint256 _nonce	onlySigner
approve	address _token address _spender uint256 _amount uint256 _nonce	onlyAdmin
signalApproveNFT	address _token address _spender uint256 _tokenId	onlyAdmin

signApproveNFT	address _token address _spender uint256 _tokenId uint256 _nonce	onlySigner
approveNFT	address _token address _spender uint256 _tokenId uint256 _nonce	onlyAdmin
signalApproveNFTs	address _token address _spender uint256[] memory _tokenIds	onlyAdmin
signApproveNFTs	address _token address _spender uint256[] memory _tokenIds uint256 _nonce	onlySigner
approveNFTs	address _token address _spender uint256[] memory _tokenIds uint256 _nonce	onlyAdmin
receiveNFTs	address _token address _sender uint256[] memory _tokenIds	onlyAdmin
signalSetAdmin	address _target address _admin	onlySigner
signSetAdmin	address _target address _admin uint256 _nonce	onlySigner
setAdmin	address _target address _admin uint256 _nonce	onlySigner
signalSetGov	address _timelock address _target address _gov	onlyAdmin
signSetGov	address _timelock address _target address _gov uint256 _nonce	onlySigner
setGov	address _timelock address _target	onlyAdmin

	address _gov uint256 _nonce	
_setPendingAction	bytes32 _action uint256 _nonce	private
_validateAction	bytes32 _action	private
_validateAuthorization	bytes32 _action	private
_clearAction	bytes32 _action uint256 _nonce	private

Timelock Contract

方法名称	方法传参	方法属性
setAdmin	address _admin	onlyTokenManager
setExternalAdmin	address _target address _admin	onlyAdmin
setContractHandler	address _handler bool _isActive	onlyAdmin
initLpManager	none	onlyAdmin
initRewardRouter	address _rewardRouter	onlyAdmin
setKeeper	address _keeper bool _isActive	onlyAdmin
setBuffer	uint256 _buffer	onlyAdmin
setMaxLeverage	address _vault uint256 _maxLeverage	onlyAdmin
setFundingRate	address _vault uint256 _fundingInterval uint256 _fundingRateFactor uint256 _stableFundingRateFactor	onlyKeeperAndAbove
setShouldToggleIsLeverageEnabled	bool _shouldToggleIsLeverageEnabled	onlyHandlerAndAbove
setMarginFeeBasisPoints	uint256 _marginFeeBasisPoints uint256 _maxMarginFeeBasisPoints	onlyHandlerAndAbove
setSwapFees	address _vault uint256 _taxBasisPoints uint256 _stableTaxBasisPoints	onlyKeeperAndAbove

	uint256 _mintBurnFeeBasisPoints uint256 _swapFeeBasisPoints uint256 _stableSwapFeeBasisPoints	
setFees	address _vault uint256 _taxBasisPoints uint256 _stableTaxBasisPoints uint256 _mintBurnFeeBasisPoints uint256 _swapFeeBasisPoints uint256 _stableSwapFeeBasisPoints uint256 _marginFeeBasisPoints uint256 _liquidationFeeUsd uint256 _minProfitTime bool _hasDynamicFees	onlyKeeperAndAbove
enableLeverage	address _vault	onlyHandlerAndAbove
disableLeverage	address _vault	onlyHandlerAndAbove
setIsLeverageEnabled	address _vault bool _isLeverageEnabled	onlyHandlerAndAbove
setTokenConfig	address _vault address _token uint256 _tokenWeight uint256 _minProfitBps uint256 _maxUsdrAmount uint256 _bufferAmount uint256 _usdrAmount	onlyKeeperAndAbove
setUsdrAmounts	address _vault address[] memory _tokens uint256[] memory _usdrAmounts	onlyKeeperAndAbove
updateUsdrSupply	uint256 usdrAmount	onlyKeeperAndAbove
setShortsTrackerAveragePriceWeight	uint256 _shortsTrackerAveragePriceWeight	onlyAdmin
setLpCooldownDuration	uint256 _cooldownDuration	onlyAdmin
removeAdmin	address _token address _account	onlyAdmin
setIsSwapEnabled	address _vault bool _isSwapEnabled	onlyKeeperAndAbove

setTier	address _referralStorage uint256 _tierId uint256 _totalRebate uint256 _discountShare	onlyKeeperAndAbove
setReferrerTier	address _referralStorage address _referrer uint256 _tierId	onlyKeeperAndAbove
govSetCodeOwner	address _referralStorage bytes32 _code address _newAccount	onlyKeeperAndAbove
setMaxGasPrice	address _vault uint256 _maxGasPrice	onlyAdmin
withdrawFees	address _vault address _token address _receiver	onlyAdmin
batchWithdrawFees	address _vault address[] memory _tokens	onlyKeeperAndAbove
setInPrivateLiquidationMode	address _vault bool _inPrivateLiquidationMode	onlyAdmin
setLiquidator	address _vault address _liquidator bool _isActive	onlyAdmin
setInPrivateTransferMode	address _token bool _inPrivateTransferMode	onlyAdmin
batchSetBonusRewards	address _vester address[] memory _accounts uint256[] memory _amounts	onlyKeeperAndAbove
transferIn	address _sender address _token uint256 _amount	onlyAdmin
signalApprove	address _token address _spender uint256 _amount	onlyAdmin
approve	address _token address _spender uint256 _amount	onlyAdmin
signalWithdrawToken	address _target address _token	onlyAdmin

	address _receiver uint256 _amount	
withdrawToken	address _target address _token address _receiver uint256 _amount	onlyAdmin
signalMint	address _token address _receiver uint256 _amount	onlyAdmin
processMint	address _token address _receiver uint256 _amount	onlyAdmin
signalSetGov	address _target address _gov	onlyAdmin
setGov	address _target address _gov	onlyAdmin
signalSetHandler	address _target address _handler bool _isActive	onlyAdmin
setHandler	address _target address _handler bool _isActive	onlyAdmin
signalSetPriceFeed	address _vault address _priceFeed	onlyAdmin
setPriceFeed	address _vault address _priceFeed	onlyAdmin
signalRedeemUsdr	address _vault address _token u int256 _amount	onlyAdmin
redeemUsdr	address _vault a ddress _token uint256 _amount	onlyAdmin
signalVaultSetTokenConfig	address _vault address _token uint256 _tokenDecimals uint256 _tokenWeight uint256 _minProfitBps uint256 _maxUsdrAmount bool _isStable	onlyAdmin

	bool _isShortable	
vaultSetTokenConfig	address _vault address _token uint256 _tokenDecimals uint256 _tokenWeight uint256 _minProfitBps uint256 _maxUsdrAmount bool _isStable bool _isShortable	onlyAdmin
cancelAction	bytes32 _action	onlyAdmin
_mint	address _token address _receiver uint256 _amount	private
_setPendingAction	bytes32 _action	private
_validateAction	bytes32 _action	private
_clearAction	bytes32 _action	private

Governable Contract

方法名称	方法传参	方法属性
setGov	address _gov	onlyGov

Reader Contract

方法名称	方法传参	方法属性
setConfig	bool _hasMaxGlobalShortSizes	onlyGov
getMaxAmountIn	IVault _vault address _tokenIn address _tokenOut	public
getAmountOut	IVault _vault address _tokenIn address _tokenOut uint256 _amountIn	public
getFeeBasisPoints	IVault _vault	public

	address _tokenIn address _tokenOut uint256 _amountIn	
getFees	address _vault address[] memory _tokens	public
getTotalStaked	address[] memory _yieldTokens	public
getStakingInfo	address _account address[] memory _yieldTrackers	public
getFundingRates	address _vault address _weth address[] memory _tokens	public
getTokenSupply	IERC20 _token address[] memory _excludedAccounts	public
getTotalBalance	IERC20 _token address[] memory _accounts	public
getTokenBalances	address _account address[] memory _tokens	public
getTokenBalancesWithSupplies	address _account address[] memory _tokens	public
getPrices	IVaultPriceFeed _priceFeed address[] memory _tokens	public
getVaultTokenInfo	address _vault address _weth uint256 _usdrAmount address[] memory _tokens	public
getFullVaultTokenInfo	address _vault address _weth uint256 _usdrAmount address[] memory _tokens	public
getPositions	address _vault address _account address[] memory _collateralTokens address[] memory _indexTokens bool[] memory _isLong	public

VaultReader Contract

方法名称	方法传参	方法属性
getVaultTokenInfoV3	address _vault address _positionManage address _weth uint256 _usdrAmount address[] memory _tokens	public view
getVaultTokenInfoV4	address _vault address _positionManager address _weth uint256 _usdrAmount address[] memory _tokens	public view

BalanceUpdater Contract

方法名称	方法传参	方法属性
updateBalance	address _vault address _token address _usdr uint256 _usdrAmount	public

BatchSender Contract

方法名称	方法传参	方法属性
setHandler	address _handler bool _isActive	onlyGov
send	IERC20 _token address[] memory _accounts uint256[] memory _amounts	onlyHandler
sendAndEmit	IERC20 _token address[] memory _accounts uint256[] memory _amounts uint256 _typeId	onlyHandler

_send	IERC20 _token address[] memory _accounts uint256[] memory _amounts uint256 _typeId	private

RewardReader Contract

方法名称	方法传参	方法属性
getDepositBalances	address _account address[] memory depositToken address[] memory _rewardTrackers	public view
getStakingInfo	address _account address[] memory _rewardTrackers	public view
getVestingInfoV2	address _account address[] memory _vesters	public view

OrderBookReader Contract

方法名称	方法传参	方法属性
getIncreaseOrders	address payable _orderBookAddress address _account uint256[] memory _indices	external
getDecreaseOrders	address payable _orderBookAddress address _account uint256[] memory _indices	external
getSwapOrders	address payable _orderBookAddress address _account uint256[] memory _indices	external

DexV3Aggregator Contract

方法名称	方法传参	方法属性
addPriceSource	address _source uint256 _weight address[] memory _path	onlyGov
removePriceSource	address _source	onlyGov
calcPrice	none	public
latestAnswer	none	public
latestTimestamp	none	public
latestRound	none	public
getAnswer	uint256	public
getTimestamp	uint256 _roundId	public
getRoundData	uint80 _roundId	external
latestRoundData	none	external
description	none	external

FastPriceEvents Contract

方法名称	方法传参	方法属性
setIsPriceFeed	address _priceFeed bool _isPriceFeed	onlyGov
emitPriceEvent	address _token uint256 _price	external

CustomV3Aggregator Contract

方法名称	方法传参	方法属性
setFastPriceFeed	address _priceFeed	onlyGov
setUpdater	address _updater bool _status	onlyGov
updateAnswer	int256 _answer	onlyUpdater
updateRoundData	uint80 _roundId int256 _answer uint256 _timestamp uint256 _startedAt	onlyUpdater
getRoundData	uint80 _roundId	external
latestRoundData	none	external
description	none	external

FastPriceFeed Contract

方法名称	方法传参	方法属性
initialize	uint256 _minAuthorizations address[] memory _signers address[] memory _updaters	onlyGov
setSigner	address _account bool _isActive	onlyGov
setUpdater	address _account bool _isActive	onlyGov
setFastPriceEvents	address _fastPriceEvents	onlyGov
setVaultPriceFeed	address _vaultPriceFeed	onlyGov
setMaxTimeDeviation	uint256 _maxTimeDeviation	onlyGov
setPriceDuration	uint256 _priceDuration	onlyGov
setMaxPriceUpdateDelay	uint256 _maxPriceUpdateDelay	onlyGov
setSpreadBasisPointsIfInactive	uint256 _spreadBasisPointsIfInactive	onlyGov

setSpreadBasisPointsIfChainError	uint256 _spreadBasisPointsIfChainError	onlyGov
setMinBlockInterval	uint256 _minBlockInterval	onlyGov
setIsSpreadEnabled	bool _isSpreadEnabled	onlyGov
setLastUpdatedAt	uint256 _lastUpdatedAt	onlyGov
setMaxDeviationBasisPoints	uint256 _maxDeviationBasisPoints	onlyGov
setMaxCumulativeDeltaDiffs	address[] memory _tokens uint256[] memory _maxCumulativeDeltaDiffs	onlyGov
setPriceDataInterval	uint256 _priceDataInterval	onlyGov
setMinAuthorizations	uint256 _minAuthorizations	onlyGov
setTokens	address[] memory _tokens uint256[] memory _tokenPrecisions	onlyGov
setPrices	address[] memory _tokens uint256[] memory _prices uint256 _timestamp	onlyUpdater
setCompactedPrices	uint256[] memory _priceBitArray uint256 _timestamp	onlyUpdater
setPricesWithBits	uint256 _priceBits uint256 _timestamp	onlyUpdater
setPricesWithBitsAndExecute	uint256 _priceBits, uint256 _timestamp uint256 _endIndexForIncreasePositions uint256 _endIndexForDecreasePositions uint256 _maxIncreasePositions, uint256 _maxDecreasePositions	onlyUpdater
disableFastPrice	none	onlySigner
enableFastPrice	none	onlySigner
getPrice	address _token uint256 _refPrice bool _maximise	public
favorFastPrice	address _token	public
getPriceData	address _token	public
_setPricesWithBits	uint256 _priceBits uint256 _timestamp	private
_setPrice	address _token uint256 _price	private

	address _vaultPriceFeed address _fastPriceEvents	
_setPriceData	address _token uint256 _refPrice uint256 _cumulativeRefDelta uint256 _cumulativeFastDelta	private
_emitPriceEvent	address _fastPriceEvents address _token uint256 _price	private
_setLastUpdatedValues	uint256 _timestamp	private

LpManager Contract

方法名称	方法传参	方法属性
setInPrivateMode	bool _inPrivateMode	onlyGov
setShortsTracker	IShortsTracker _shortsTracker	onlyGov
setShortsTrackerAveragePriceWeight	uint256 _shortsTrackerAveragePriceWeight	onlyGov
setHandler	address _handler bool _isActive	onlyGov
setCooldownDuration	uint256 _cooldownDuration	onlyGov
setAumAdjustment	uint256 _aumAddition uint256 _aumDeduction	onlyGov
addLiquidity	address _token uint256 _amount uint256 _minUsdr uint256 _minLp	external
addLiquidityForAccount	address _fundingAccount address _account address _token uint256 _amount uint256 _minUsdr uint256 _minLp	external
removeLiquidity	address _tokenOut	external

	uint256 _lpAmount uint256 _minOut address _receiver	
removeLiquidityForAccount	address _account address _tokenOut uint256 _lpAmount uint256 _minOut address _receiver	external
getPrice	bool _maximise	external
getAums	none	public
getAumInUsdr	bool maximise	public
getAum	bool maximise	public
getGlobalShortDelta	address _token uint256 _price uint256 _size	public
getGlobalShortAveragePrice	address _token	public
_addLiquidity	address _fundingAccount address _account address _token uint256 _amount uint256 _minUsdr uint256 _minLp	private
_removeLiquidity	address _account address _tokenOut uint256 _lpAmount uint256 _minOut address _receiver	private
_validateHandler	none	private

VaultWrapper Contract

方法名称	方法传参	方法属性
setShouldToggleIsLeverageEnabled	bool _shouldToggleIsLeverageEnabled	onlyGov
setMarginFeeBasisPoints	uint256 _marginFeeBasisPoints uint256 _maxMarginFeeBasisPoints	onlyGov
enableLeverage	address _vault	external
disableLeverage	address _vault	external

PositionRouter Contract

方法名称	方法传参	方法属性
setPositionKeeper	address _account bool _isActive	onlyAdmin
setCallbackGasLimit	uint256 _callbackGasLimit	onlyAdmin
setMinExecutionFee	uint256 _minExecutionFee	onlyAdmin
setIsLeverageEnabled	bool _isLeverageEnabled	onlyAdmin
setDelayValues	uint256 _minBlockDelayKeeper uint256 _minTimeDelayPublic uint256 _maxTimeDelay	onlyAdmin
setRequestKeysStartValues	uint256 _increasePositionRequestKeysStart uint256 _decreasePositionRequestKeysStart	onlyAdmin
executeIncreasePositions	uint256 _endIndex address payable _executionFeeReceiver	onlyPositionKeeper
executeDecreasePositions	uint256 _endIndex address payable _executionFeeReceiver	onlyPositionKeeper
createIncreasePosition	address[] memory _path address _indexToken uint256 _amountIn uint256 _minOut uint256 _sizeDelta	external

	bool _isLong uint256 _acceptablePrice uint256 _executionFee bytes32 _referralCode address _callbackTarget	
createIncreasePositionETH	address[] memory _path address _indexToken uint256 _minOut uint256 _sizeDelta bool _isLong uint256 _acceptablePrice uint256 _executionFee bytes32 _referralCode address _callbackTarget	external
createDecreasePosition	address[] memory _path address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _acceptablePrice uint256 _minOut uint256 _executionFee bool _withdrawETH address _callbackTarget	external
getRequestQueueLengths	none	external
executeIncreasePosition	bytes32 _key address payable _executionFeeReceiver	public
cancelIncreasePosition	bytes32 _key address payable _executionFeeReceiver	public
executeDecreasePosition	bytes32 _key address payable _executionFeeReceiver	public
cancelDecreasePosition	bytes32 _key address payable _executionFeeReceiver	public
getRequestKey	address _account uint256 _index	public
getIncreasePositionRequestPath	bytes32 _key	public

getDecreasePositionRequestPath	bytes32 _key	public
_setTraderReferralCode	bytes32 _referralCode	internal
_validateExecution	uint256 _positionBlockNumber uint256 _positionBlockTime address _account	internal
_validateCancellation	uint256 _positionBlockNumber uint256 _positionBlockTime address _account	internal
_createIncreasePosition	address _account address[] memory _path address _indexToken uint256 _amountIn uint256 _minOut uint256 _sizeDelta bool _isLong uint256 _acceptablePrice uint256 _executionFee bool _hasCollateralInETH address _callbackTarget	internal
_storeIncreasePositionRequest	IncreasePositionRequest memory _request	internal
_storeDecreasePositionRequest	DecreasePositionRequest memory _request	internal
_createDecreasePosition	address _account address[] memory _path address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _acceptablePrice uint256 _minOut uint256 _executionFee bool _withdrawETH address _callbackTarget	internal
_callRequestCallback	address _callbackTarget bytes32 _key	internal

	bool _wasExecuted bool _isIncrease	

VaultPriceFeed Contract

方法名称	方法传参	方法属性
setGov	address _gov	onlyGov
setChainlinkFlags	address _chainlinkFlags	onlyGov
setAdjustment	address _token bool _isAdditive uint256 _adjustmentBps	onlyGov
setUseV2Pricing	bool _useV2Pricing	onlyGov
setIsAmmEnabled	bool _isEnabled	onlyGov
setIsSecondaryPriceEnabled	bool _isEnabled	onlyGov
setSecondaryPriceFeed	address _secondaryPriceFeed	onlyGov
setTokens	address _btc address _eth address _bnb	onlyGov
setPairs	address _bnbBusd address _ethBnb address _btcBnb	onlyGov
setSpreadBasisPoints	address _token uint256 _spreadBasisPoints	onlyGov
setSpreadThresholdBasisPoints	uint256 _spreadThresholdBasisPoints	onlyGov
setFavorPrimaryPrice	bool _favorPrimaryPrice	onlyGov
setPriceSampleSpace	uint256 _priceSampleSpace	onlyGov
setMaxStrictPriceDeviation	uint256 _maxStrictPriceDeviation	onlyGov
setTokenConfig	address _token address _priceFeed uint256 _priceDecimals bool _isStrictStable	onlyGov
getPrice	address _token bool _maximise	public override view

	bool _includeAmmPrice bool /* _useSwapPricing */	
getPriceV1	address _token bool _maximise bool _includeAmmPrice	public view
getPriceV2	address _token bool _maximise bool _includeAmmPrice	public view
getAmmPriceV2	address _token bool _maximise uint256 _primaryPrice	public view
getLatestPrimaryPrice	address _token	public override view
getPrimaryPrice	address _token bool _maximise	public override view
getSecondaryPrice	address _token uint256 _referencePrice bool _maximise	public view
getAmmPrice	address _token	public override view
getPairPrice	address _pair bool _divByReserve0	public view

PositionManager Contract

方法名称	方法传参	方法属性
setOrderKeeper	address _account bool _isActive	onlyAdmin
setLiquidator	address _account bool _isActive	onlyAdmin
setPartner	address _account bool _isActive	onlyAdmin
setShouldValidateIncreaseOrder	bool _shouldValidateIncreaseOrder	onlyAdmin
increasePosition	address[] memory _path	onlyPartners

	address _indexToken uint256 _amountIn uint256 _minOut uint256 _sizeDelta bool _isLong uint256 _price	
increasePositionETH	address[] memory _path address _indexToken uint256 _minOut uint256 _sizeDelta bool _isLong uint256 _price	onlyPartners
decreasePosition	address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _price	onlyPartners
decreasePositionETH	address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _price	onlyPartner
decreasePositionAndSwap	address[] memory _path address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _price uint256 _minOut	onlyPartner
decreasePositionAndSwapETH	address[] memory _path address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong	onlyPartner

	address _receiver uint256 _price uint256 _minOut	
liquidatePosition	address _account address _collateralToken address _indexToken bool _isLong,address _feeReceiver	onlyLiquidator
executeSwapOrder	address _account uint256 _orderIndex address payable _feeReceiver	onlyOrderKeeper
executeIncreaseOrder	address _account uint256 _orderIndex address payable _feeReceiver	onlyOrderKeeper
executeDecreaseOrder	address _account uint256 _orderIndex address payable _feeReceiver	onlyOrderKeeper
_validateIncreaseOrder	address _account uint256 _orderIndex	internal view

ShortsTracker Contract

方法名称	方法传参	方法属性
setHandler	address _handler bool _isActive	onlyGov
_setGlobalShortAveragePrice	address _token uint256 _averagePrice	internal
setIsGlobalShortDataReady	bool value	onlyGov
updateGlobalShortData	address _account address _collateralToken address _indexToken bool _isLong uint256 _sizeDelta uint256 _markPrice bool _isIncrease	onlyHandler
getGlobalShortDelta	address _token	public
setInitData	address[] calldata _tokens	onlyGov

	uint256[] calldata _averagePrices	
getNextGlobalShortData	address _account address _collateralToken address _indexToken uint256 _nextPrice uint256 _sizeDelta bool _isIncrease	public
getRealisedPnl	address _account address _collateralToken address _indexToken uint256 _sizeDelta bool _isIncrease	public
_getNextGlobalAveragePrice	uint256 _averagePrice uint256 _nextPrice uint256 _nextSize uint256 _delta int256 _realisedPnl	public
_getNextDelta	uint256 _delta uint256 _averagePrice uint256 _nextPrice int256 _realisedPnl	internal

OrderBook Contract

方法名称	方法传参	方法属性
initialize	address _router address _vault address _weth address _usdr uint256 _minExecutionFee uint256 _minPurchaseTokenAmountUsd	onlyGov
setMinExecutionFee	uint256 _minExecutionFee	onlyGov
setMinPurchaseTokenAmountUsd	uint256 _minPurchaseTokenAmountUsd	onlyGov
setGov	address _gov	onlyGov
getSwapOrder	address _account uint256 _orderIndex	public
createSwapOrder	address[] memory _path	external

	uint256 _amountIn uint256 _minOut uint256 _triggerRatio bool _triggerAboveThreshold uint256 _executionFee bool _shouldWrap bool _shouldUnwrap	
_createSwapOrder	address _account address[] memory _path uint256 _amountIn uint256 _minOut uint256 _triggerRatio bool _triggerAboveThreshold bool _shouldUnwrap uint256 _executionFee	private
cancelMultiple	uint256[] memory _swapOrderIndexes uint256[] memory _increaseOrderIndexes uint256[] memory _decreaseOrderIndexes	external
cancelSwapOrder	uint256 _orderIndex	public
getUsdrMinPrice	address _otherToken	public
validateSwapOrderPriceWithTriggerAboveThreshold	address[] memory _path uint256 _triggerRatio	public
updateSwapOrder	uint256 _orderIndex uint256 _minOut uint256 _triggerRatio bool _triggerAboveThreshold	external
executeSwapOrder	address _account uint256 _orderIndex address payable _feeReceiver	external
validatePositionOrderPrice	bool _triggerAboveThreshold uint256 _triggerPrice address _indexToken bool _maximizePrice bool _raise	public
getDecreaseOrder	address _account uint256 _orderIndex	public
getIncreaseOrder	address _account uint256 _orderIndex	public

createIncreaseOrder	address[] memory _path uint256 _amountIn address _indexToken uint256 _minOut uint256 _sizeDelta address _collateralToken bool _isLong uint256 _triggerPrice bool _triggerAboveThreshold uint256 _executionFee bool _shouldWrap	external
_createIncreaseOrder	address _account address _purchaseToken uint256 _purchaseTokenAmount address _collateralToken address _indexToken uint256 _sizeDelta bool _isLong uint256 _triggerPrice bool _triggerAboveThreshold uint256 _executionFee	private
updateIncreaseOrder	uint256 _orderIndex uint256 _sizeDelta uint256 _triggerPrice bool _triggerAboveThreshold	external
cancelIncreaseOrder	uint256 _orderIndex	public
executeIncreaseOrder	address _address uint256 _orderIndex address payable _feeReceiver	external
createDecreaseOrder	address _indexToken uint256 _sizeDelta address _collateralToken uint256 _collateralDelta bool _isLong uint256 _triggerPrice bool _triggerAboveThreshold	external
_createDecreaseOrder	address _account address _collateralToken uint256 _collateralDelta	private

	address _indexToken uint256 _sizeDelta bool _isLong uint256 _triggerPrice bool _triggerAboveThreshold	
executeDecreaseOrder	address _address uint256 _orderIndex address payable _feeReceiver	external
cancelDecreaseOrder	uint256 _orderIndex	public
updateDecreaseOrder	uint256 _orderIndex uint256 _collateralDelta uint256 _sizeDelta uint256 _triggerPrice bool _triggerAboveThreshold	external
_transferInETH	none	private
_transferOutETH	uint256 _amountOut address payable _receiver	private
_swap	none	
_vaultSwap	address _tokenIn address _tokenOut uint256 _minOut address _receiver	private

Vault Contract

方法名称	方法传参	方法属性
initialize	address _router address _usdr address _priceFeed uint256 _liquidationFeeUsd uint256 _fundingRateFactor uint256 _stableFundingRateFactor	external
allWhitelistedTokensLength	none	external
setInManagerMode	bool _inManagerMode	external

setManager	address _manager bool _isManager	external
setInPrivateLiquidationMode	bool _inPrivateLiquidationMode	external
setLiquidator	address _liquidator bool _isActive	external
setIsSwapEnabled	bool _isSwapEnabled	external
setIsLeverageEnabled	bool _isLeverageEnabled	external
setMaxGasPrice	uint256 _maxGasPrice	external
setWrapper	address _wrapper	external
setGov	address _gov	external
setPriceFeed	address _priceFeed	external
setMaxLeverage	uint256 _maxLeverage	external
setBufferAmount	address _token uint256 _amount	external
setFees	uint256 _taxBasisPoints uint256 _stableTaxBasisPoints uint256 _mintBurnFeeBasisPoints uint256 _swapFeeBasisPoints uint256 _stableSwapFeeBasisPoints uint256 _marginFeeBasisPoints uint256 _liquidationFeeUsd uint256 _minProfitTime bool _hasDynamicFees	external
setFundingRate	uint256 _fundingInterval uint256 _fundingRateFactor uint256 _stableFundingRateFactor	external
setTokenConfig	address _token uint256 _tokenDecimals uint256 _tokenWeight uint256 _minProfitBps uint256 _maxUsdrAmount bool _isStable bool _isShortable	external
clearTokenConfig	address _token	external
withdrawFees	address _token address _receiver	external
addRouter	address _router	external

removeRouter	address _router	external
setUsdrAmount	address _token uint256 _amount	external
upgradeVault	address _newVault address _token uint256 _amount	external
directPoolDeposit	address _token	external
buyUSDR	address _token address _receiver	external
sellUSDR	address _token address _receiver	external
swap	address _tokenIn address _tokenOut address _receiver	external
increasePosition	address _account address _collateralToken address _indexToken uint256 _sizeDelta bool _isLong	external
decreasePosition	address _account address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver	external
_decreasePosition	address _account address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver	private
liquidatePosition	address _account address _collateralToken address _indexToken b ool _isLong address _feeReceiver	external
validateLiquidation	address _account	public

	address _collateralToken address _indexToken bool _isLong bool _raise	
getMaxPrice	address _token	public
getMinPrice	address _token	public
getRedemptionAmount	address _token uint256 _usdrAmount	public
getRedemptionCollateral	address _token	public
getRedemptionCollateralUsd	address _token	public
adjustForDecimals	uint256 _amount address _tokenDiv address _tokenMul	public
tokenToUsdMin	address _token uint256 _tokenAmount	public
usdToTokenMax	address _token uint256 _usdAmount	public
usdToTokenMin	address _token uint256 _usdAmount	public
usdToToken	address _token uint256 _usdAmount uint256 _price	public
getPosition	address _account address _collateralToken address _indexToken bool _isLong	public
getPositionKey	address _account address _collateralToken address _indexToken bool _isLong	public
updateCumulativeFundingRate	address _token	public
getNextFundingRate	address _token	public
getUtilisation	address _token	public
getPositionLeverage	address _account address _collateralToken address _indexToken	public

	bool _isLong	
getNextAveragePrice	address _indexToken uint256 _size uint256 _averagePrice bool _isLong uint256 _nextPrice uint256 _sizeDelta uint256 _lastIncreasedTime	public
getNextGlobalShortAveragePrice	address _indexToken uint256 _nextPrice uint256 _sizeDelta	public
getGlobalShortDelta	address _token	public
getPositionDelta	address _account address _collateralToken address _indexToken bool _isLong	public
getDelta	address _indexToken uint256 _size uint256 _averagePrice bool _isLong uint256 _lastIncreasedTime	public
getFundingFee	address _token uint256 _size uint256 _entryFundingRate	public
getPositionFee	uint256 _sizeDelta	public
getFeeBasisPoints	address _token uint256 _usdrDelta uint256 _feeBasisPoints uint256 _taxBasisPoints bool _increment	public
getTargetUsdrAmount	address _token	public
_reduceCollateral	address _account address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong	private
_validatePosition	uint256 _size uint256 _collateral	private

_validateRouter	address _account	private
_validateTokens	address _collateralToken address _indexToken bool _isLong	private
_collectSwapFees	address _token uint256 _amount uint256 _feeBasisPoints	private
_collectMarginFees	address _token uint256 _sizeDelta uint256 _size uint256 _entryFundingRate	private
_transferIn	address _token	private
_transferOut	address _token uint256 _amount address _receiver	private
_updateTokenBalance	address _token	private
_increasePoolAmount	address _token uint256 _amount	private
_decreasePoolAmount	address _token uint256 _amount	private
_validateBufferAmount	address _token	private
_increaseUsdrAmount	address _token uint256 _amount	private
_decreaseUsdrAmount	address _token uint256 _amount	private
_increaseReservedAmount	address _token uint256 _amount	private
_decreaseReservedAmount	address _token uint256 _amount	private
_increaseGuaranteedUsd	address _token uint256 _usdAmount	private
_decreaseGuaranteedUsd	address _token uint256 _usdAmount	private
_decreaseGlobalShortSize	address _token uint256 _amount	private
_onlyGov	none	private
_validateManager	none	private

_validateGasPrice	none	private
_onlyGovOrWrapper	none	private

Router Contract

方法名称	方法传参	方法属性
setGov	address _gov	external
addPlugin	address _plugin	external
removePlugin	address _plugin	external
approvePlugin	address _plugin	external
denyPlugin	address _plugin	external
pluginTransfer	address _token address _account address _receiver uint256 _amount	external
pluginIncreasePosition	address _account address _collateralToken address _indexToken uint256 _sizeDelta bool _isLong	external
pluginDecreasePosition	address _account address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver	external
directPoolDeposit	address _token uint256 _amount	external
swap	address[] memory _path uint256 _amountIn uint256 _minOut address _receiver	public
swapETHToTokens	address[] memory _path uint256 _minOut	external

	address _receiver	
swapTokensToETH	address[] memory _path uint256 _amountIn uint256 _minOut address payable _receiver	external
increasePosition	address[] memory _path address _indexToken uint256 _amountIn uint256 _minOut uint256 _sizeDelta bool _isLong uint256 _price	external
increasePositionETH	address[] memory _path address _indexToken uint256 _minOut uint256 _sizeDelta bool _isLong uint256 _price	external
decreasePosition	address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _price	external
decreasePositionETH	address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address payable _receiver uint256 _price	external
decreasePositionAndSwap	address[] memory _path address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _price	external

	uint256 _minOut	
decreasePositionAndSwapETH	address[] memory _path address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address payable _receiver uint256 _price uint256 _minOut	external
_increasePosition	address _collateralToken address _indexToken uint256 _sizeDelta bool _isLong uint256 _price	private
_decreasePosition	address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _price	private
_transferETHToVault	none	private
_transferOutETH	uint256 _amountOut address payable _receiver	private
_swap	address[] memory _path uint256 _minOut address _receiver	private
_vaultSwap	address _tokenIn address _tokenOut uint256 _minOut address _receiver	private
_sender	none	private
_validatePlugin	address _account	private

BasePositionManager Contract

方法名称	方法传参	方法属性
setAdmin	address _admin	onlyGov
setDepositFee	uint256 _depositFee	onlyAdmin
setIncreasePositionBufferBps	uint256 _increasePositionBufferBps	onlyAdmin
setReferralStorage	address _referralStorage	onlyAdmin
setMaxGlobalSizes	address[] memory _tokens uint256[] memory _longSizes uint256[] memory _shortSizes	onlyAdmin
withdrawFees	address _token address _receiver	onlyAdmin
approve	address _token address _spender uint256 _amount	onlyGov
sendValue	addresspayable _receiver uint256 _amount	onlyGov
_validateMaxGlobalSize	address _indexToken bool _isLong uint256 _sizeDelta	internal
_increasePosition	address _account address _collateralToken address _indexToken uint256 _sizeDelta bool _isLong uint256 _price	internal
_decreasePosition	address _account address _collateralToken address _indexToken uint256 _collateralDelta uint256 _sizeDelta bool _isLong address _receiver uint256 _price	internal
_emitIncreasePositionReferral	address _account uint256 _sizeDelta	internal
_emitDecreasePositionReferral	address _account uint256 _sizeDelta	internal
_swap	address[] memory _path	internal

	uint256 _minOut address _receiver	
_vaultSwap	address _tokenIn address _tokenOut uint256 _minOut address _receiver	internal
_transferInETH	none	internal
_transferOutETHWithGasLimitIgnoreFail	uint256 _amountOut address payable _receiver	internal
_collectFees	address _account address[] memory _path uint256 _amountIn address _indexToken bool _isLong uint256 _sizeDelta	internal
_shouldDeductFee	address _account address[] memory _path uint256 _amountIn address _indexToken bool _isLong uint256 _sizeDelta	internal

RewardTracker Contract

方法名称	方法传参	方法属性
initialize	address[] memory _depositTokens address _distributor	onlyGov
setDepositToken	address _depositToken bool _isDepositToken	onlyGov
setInPrivateTransferMode	bool _inPrivateTransferMode	onlyGov
setInPrivateStakingMode	bool _inPrivateStakingMode	onlyGov
setInPrivateClaimingMode	bool _inPrivateClaimingMode	onlyGov

setHandler	address _handler bool _isActive	onlyGov
withdrawToken	address _token address _account uint256 _amount	onlyGov
balanceOf	address _account	external
stake	address _depositToken uint256 _amount	external
stakeForAccount	address _fundingAccount address _account address _depositToken uint256 _amount	external
unstake	address _depositToken uint256 _amount	external
unstakeForAccount	address _account address _depositToken uint256 _amount address _receiver	external
transfer	address _recipient uint256 _amount	external
allowance	address _owner address _spender	external
approve	address _spender uint256 _amount	external
transferFrom	address _sender address _recipient uint256 _amount	external
tokensPerInterval	none	external
updateRewards	none	external
claim	address _receiver	external
claimForAccount	address _account address _receiver	external
claimable	address _account	public
rewardToken	none	public
_claim	address _account address _receiver	private
_mint	address _account uint256 _amount	internal

_burn	address _account uint256 _amount	internal
_transfer	address _sender address _recipient uint256 _amount	private
_approve	address _owner address _spender uint256 _amount	private
_validateHandler	none	private
_stake	address _fundingAccount address _account address _depositToken uint256 _amount	private
_unstake	address _account address _depositToken uint256 _amount address _receiver	private
_updateRewards	address _account	private

RewardRouterV1 Contract

方法名称	方法传参	方法属性
initialize	address _weth address _lp address _feeLpTracker address _lpManager	onlyGov
withdrawToken	address _token address _account uint256 _amount	onlyGov
mintAndStakeLp	address _token uint256 _amount uint256 _minUsdr uint256 _minLp	external
mintAndStakeLpETH	uint256 _minUsdr uint256 _minLp	external

unstakeAndRedeemLp	address _tokenOut uint256 _lpAmount uint256 _minOut address _receiver	external
unstakeAndRedeemLpETH	uint256 _lpAmount uint256 _minOut addresspayable _receiver	external
claim	none	external
claimFees	none	external
handleRewards	bool _shouldConvertWethToEth	external
signalTransfer	address _receiver	external
acceptTransfer	address _sender	external
_validateReceiver	address _receiver	private

RewardDistributor Contract

方法名称	方法传参	方法属性
setAdmin	address _admin	onlyGov
withdrawToken	address _token address _account uint256 _amount	onlyGov
updateLastDistributionTime	none	onlyAdmin
setTokensPerInterval	uint256 _amount	onlyAdmin
pendingRewards	none	public
distribute	none	external

ReferralStorage Contract

方法名称	方法传参	方法属性
setHandler	address _handler bool _isActive	onlyGov

setTier	uint256 _tierId uint256 _totalRebate uint256 _discountShare	onlyGov
setReferrerTier	address _referrer uint256 _tierId	onlyGov
setReferrerDiscountShare	uint256 _discountShare	external
setTraderReferralCode	address _account bytes32 _code	onlyHandler
setTraderReferralCodeByUser	bytes32 _code	external
registerCode	bytes32 _code	external
setCodeOwner	bytes32 _code address _newAccount	external
govSetCodeOwner	bytes32 _code address _newAccount	onlyGov
getTraderReferralInfo	address _account	external
_setTraderReferralCode	address _account bytes32 _code	private

ReferralReader Contract

方法名称	方法传参	方法属性
getCodeOwners	IReferralStorage _referralStorage bytes32[] memory _codes	public

4.审计详情

4.1 风险分布

风险名称	风险级别	修复状态
管理员权限	低	已确认
同地址判断	提示	已确认
逻辑设计缺陷	提示	部分已修复
冗余代码	提示	部分已修复
重入攻击	无	正常
变量更新问题	无	正常
整数溢出	无	正常
浮点数和数值精度	无	正常
默认可见性	无	正常
tx.origin 身份认证	无	正常
错误的构造函数	无	正常
未验证返回值	无	正常
不安全的随机数	无	正常
时间戳依赖	无	正常
交易顺序依赖	无	正常
Delegatecall 函数调用	无	正常
Call 函数调用	无	正常
拒绝服务	无	正常
假充值漏洞	无	正常
短地址攻击漏洞	无	正常
未初始化的存储指针	无	正常
冻结账户绕过	无	正常
合约调用者未初始化	无	正常

4.2 风险审计详情

4.2.1 管理员权限

- **风险描述**

upgradeVault 函数为 gov 权限调用，当 gov 特权角色为 EOA 地址时可以直接将 vault 合约中资金转出，建议使用 TimeLock 合约对此函数限制操作。

```
  function upgradeVault(address _newVault, address _token, uint256 _amo
unt) external {
    _onlyGov();
    IERC20(_token).safeTransfer(_newVault, _amount);
}
```

- **安全建议**

合约配置相关以及高权限转账的重要函数尽量使用多签或时间锁控制，避免使用 EOA 地址进行管理。

- **修复状态**

Rollup.Finance 项目方已确认。

4.2.2 同地址判断

- **风险描述**

项目中有多个合约存在 _vaultSwap 函数，其都调用的时 vault 合约的函数,由于 buyUSDR 函数以及 sellUSDR 函数中均未对 _token 参数校验是否等于 USDR 代币地址，可能存在用 USDR 执行交易以获得 USDR 的情况。

```
function _vaultSwap(address _tokenIn, address _tokenOut, uint256
_minOut, address _receiver) private returns (uint256) {
    uint256 amountOut;
    if (_tokenOut == rusd) { // buyRUSD
        amountOut = IVault(vault).buyRUSD(_tokenIn, _receiver);
    } else if (_tokenIn == rusd) { // sellRUSD
        amountOut = IVault(vault).sellRUSD(_tokenOut, _receiver);
    } else { // swap
        amountOut = IVault(vault).swap(_tokenIn, _tokenOut, _receiver);
    }
    require(amountOut >= _minOut, "Router: insufficient amountOut");
    return amountOut;
```

```
}
function sellUSDR(address _token, address _receiver) external override
nonReentrant returns (uint256) {
    _validateManager();
    require(whitelistedTokens[_token], "19");
    useSwapPricing = true;
    uint256 usdrAmount = _transferIn(usdr);
    require(usdrAmount > 0, "20");
    updateCumulativeFundingRate(_token);
    uint256 redemptionAmount = getRedemptionAmount(_token, usdrAmount);
    require(redemptionAmount > 0, "21");
    _decreaseUsdrAmount(_token, usdrAmount);
    _decreasePoolAmount(_token, redemptionAmount);
    IUSDR(usdr).burn(address(this), usdrAmount);
    _updateTokenBalance(usdr);
    uint256 feeBasisPoints = getFeeBasisPoints(_token, usdrAmount,
mintBurnFeeBasisPoints, taxBasisPoints, false);
    uint256 amountOut = _collectSwapFees(_token, redemptionAmount,
feeBasisPoints);
    require(amountOut > 0, "22");
    _transferOut(_token, amountOut, _receiver);
    emit SellUSDR(_receiver, _token, usdrAmount, amountOut,
feeBasisPoints);
    useSwapPricing = false;
    return amountOut;
}
function buyUSDR(address _token, address _receiver) external override
nonReentrant returns (uint256) {
    _validateManager();
    require(whitelistedTokens[_token], "16");
    useSwapPricing = true;
    uint256 tokenAmount = _transferIn(_token);
    require(tokenAmount > 0, "17");
    updateCumulativeFundingRate(_token);
    uint256 price = getMinPrice(_token);
    uint256 usdrAmount = tokenAmount.mul(price).div(PRICE_PRECISION);
    usdrAmount = adjustForDecimals(usdrAmount, _token, usdr);
    require(usdrAmount > 0, "18");
    uint256 feeBasisPoints = getFeeBasisPoints(_token, usdrAmount,
mintBurnFeeBasisPoints, taxBasisPoints, true);
    uint256 amountAfterFees = _collectSwapFees(_token, tokenAmount,
feeBasisPoints);
    uint256 mintAmount =
amountAfterFees.mul(price).div(PRICE_PRECISION);
    mintAmount = adjustForDecimals(mintAmount, _token, usdr);
    _increaseUsdrAmount(_token, mintAmount);
    _increasePoolAmount(_token, amountAfterFees);
    IUSDR(usdr).mint(_receiver, mintAmount);
    emit BuyUSDR(_receiver, _token, tokenAmount, mintAmount,
feeBasisPoints);
    useSwapPricing = false;
    return mintAmount;
}
```

- **安全建议**

为 buyUSDR 与 sellUSDR 函数添加对买卖的代币限制，禁止使用相同代币兑换相同代币。

- **修复状态**

Rollup.Finance 项目方已确认。

4.2.3 逻辑设计缺陷

- **风险描述**

1. depositFee 变量未限制最大值

   风险等级：提示

   depositFee 变量在 _collectFees 方法中被用作费用的计算，BASIS_POINTS_DIVISOR 变量恒为 10000，但当 depositFee 变量大于 10000 时，BASIS_POINTS_DIVISOR.sub(depositFee) 计算为负值，会出现计算错误，由于该变量由管理员设置，并且未限制其最大值。

```
function setDepositFee(uint256 _depositFee) external onlyAdmin {
    depositFee = _depositFee;
    emit SetDepositFee(_depositFee);
}
function _collectFees(
    address _account,
    address[] memory _path,
    uint256 _amountIn,
    address _indexToken,
    bool _isLong,
    uint256 _sizeDelta
) internal returns (uint256) {
    bool shouldDeductFee = _shouldDeductFee(
        _account,
        _path,
        _amountIn,
        _indexToken,
        _isLong,
        _sizeDelta
    );
    if (shouldDeductFee) {
        uint256 afterFeeAmount =
_amountIn.mul(BASIS_POINTS_DIVISOR.sub(depositFee)).div(BASIS_POINT
S_DIVISOR);
        uint256 feeAmount = _amountIn.sub(afterFeeAmount);
        address feeToken = _path[_path.length - 1];
        feeReserves[feeToken] =
feeReserves[feeToken].add(feeAmount);
        return afterFeeAmount;
    }
    return _amountIn;
}
```

2. 最新添加流动性使得之前的流动性证明代币冷却

风险等级：提示

假如用户通过 addLiquidity 和 addLiquidityETH 添加过流动性， 由于流动性资金存在全局变量的冷却时间， 之后当用户添加一笔新的流动性时， 之前的流动性证明代币也进行了冷却。

```
function _removeLiquidity(address _account, address _tokenOut,
uint256 _lpAmount, uint256 _minOut, address _receiver) private
returns (uint256) {
    require(_lpAmount > 0, "invalid _lpAmount");
    require(lastAddedAt[_account].add(cooldownDuration) <=
block.timestamp, "cooldown duration not yet passed");
    // calculate aum before sellUSDR
    uint256 aumInUsdr = getAumInUsdr(false);
    uint256 lpSupply = IERC20(lp).totalSupply();
    uint256 usdrAmount = _lpAmount.mul(aumInUsdr).div(lpSupply);
    uint256 usdrBalance = IERC20(usdr).balanceOf(address(this));
    if (usdrAmount > usdrBalance) {
        IUSDR(usdr).mint(address(this),
usdrAmount.sub(usdrBalance));
    }
    IMintable(lp).burn(_account, _lpAmount);
    IERC20(usdr).transfer(address(vault), usdrAmount);
    uint256 amountOut = vault.sellUSDR(_tokenOut, _receiver);
    require(amountOut >= _minOut, "insufficient output");
    emit RemoveLiquidity(_account, _tokenOut, _lpAmount, aumInUsdr,
lpSupply, usdrAmount, amountOut);
    return amountOut;
}
```

3. minExecutionFee 为 0 时可以无手续费调用 createIncreasePosition 方法

createIncreasePosition 和 createIncreasePositionETH 方法调用时会检查 _executionFee 和 path， 当 minExecutionFee 变量为 0 零， 所有条件都可绕过， 从而达到 0 手续费调用。

```
function setMinExecutionFee(uint256 _minExecutionFee) external
onlyAdmin {
    minExecutionFee = _minExecutionFee;
```

```
        emit SetMinExecutionFee(_minExecutionFee);
    }
```

4. gov 可能为 address(0)，建议增加 0 地址判断。

   通过多签设置的 gov 地址未对新地址进行校验，存在可能为 0 地址的风险

```
function signalSetGov(address _target, address _gov) external ove
rride onlyAdmin {
    bytes32 action = keccak256(abi.encodePacked("setGov", _targe
t, _gov));
    _setPendingAction(action);
    emit SignalSetGov(_target, _gov, action);
}
```

- **安全建议**

1. 建议 depositFee 时，增加判断条件，避免 depositFee 大于 10000 时，影响项目正常运行。
2. 为每一笔流动性添加设置独立的冷却时间，则后续添加流动性则不会覆盖之前的质押冷却时间。
3. 添加对更新该参数的函数处添加对值的校验确保其不等于 0。
4. 添加对 0 地址校验。

- **修复状态**

1. Rollup.Finance 项目方已确认。
2. Rollup.Finance 项目方已确认。
3. Rollup.Finance 项目方已确认。
4. Rollup.Finance 项目方修复。

4.2.21 代码冗余

- **漏洞描述**

1. V3 与 V4 代码重合率过高, 函数具体功能几乎一样，只存在一个参数差异。

```
function getVaultTokenInfoV3(address _vault, address _positionManager,
address _weth, uint256 _usdrAmount, address[] memory _tokens) public
view returns (uint256[] memory) {
    uint256 propsLength = 14;
    IVault vault = IVault(_vault);
    IVaultPriceFeed priceFeed = IVaultPriceFeed(vault.priceFeed());
    IBasePositionManager positionManager =
IBasePositionManager(_positionManager);
    uint256[] memory amounts = new uint256[](_tokens.length *
propsLength);
    for (uint256 i = 0; i < _tokens.length; i++) {
        address token = _tokens[i];
        if (token == address(0)) {
            token = _weth;
        }
        amounts[i * propsLength] = vault.poolAmounts(token);
        amounts[i * propsLength + 1] = vault.reservedAmounts(token);
        amounts[i * propsLength + 2] = vault.usdrAmounts(token);
        amounts[i * propsLength + 3] = vault.getRedemptionAmount(token,
_usdrAmount);
        amounts[i * propsLength + 4] = vault.tokenWeights(token);
        amounts[i * propsLength + 5] = vault.bufferAmounts(token);
        amounts[i * propsLength + 6] = vault.maxUsdrAmounts(token);
        amounts[i * propsLength + 7] = vault.globalShortSizes(token);
        amounts[i * propsLength + 8] =
positionManager.maxGlobalShortSizes(token);
        amounts[i * propsLength + 9] = vault.getMinPrice(token);
        amounts[i * propsLength + 10] = vault.getMaxPrice(token);
        amounts[i * propsLength + 11] = vault.guaranteedUsd(token);
        amounts[i * propsLength + 12] =
priceFeed.getPrimaryPrice(token, false);
        amounts[i * propsLength + 13] =
priceFeed.getPrimaryPrice(token, true);
    }
    return amounts;
}
function getVaultTokenInfoV4(address _vault, address _positionManager,
address _weth, uint256 _usdrAmount, address[] memory _tokens) public
view returns (uint256[] memory) {
    uint256 propsLength = 15;
    IVault vault = IVault(_vault);
    IVaultPriceFeed priceFeed = IVaultPriceFeed(vault.priceFeed());
    IBasePositionManager positionManager =
IBasePositionManager(_positionManager);
    uint256[] memory amounts = new uint256[](_tokens.length *
propsLength);
    for (uint256 i = 0; i < _tokens.length; i++) {
        address token = _tokens[i];
        if (token == address(0)) {
            token = _weth;
        }
        amounts[i * propsLength] = vault.poolAmounts(token);
        amounts[i * propsLength + 1] = vault.reservedAmounts(token);
        amounts[i * propsLength + 2] = vault.usdrAmounts(token);
        amounts[i * propsLength + 3] = vault.getRedemptionAmount(token,
_usdrAmount);
        amounts[i * propsLength + 4] = vault.tokenWeights(token);
        amounts[i * propsLength + 5] = vault.bufferAmounts(token);
        amounts[i * propsLength + 6] = vault.maxUsdrAmounts(token);
        amounts[i * propsLength + 7] = vault.globalShortSizes(token);
        amounts[i * propsLength + 8] =
positionManager.maxGlobalShortSizes(token);
        amounts[i * propsLength + 9] =
positionManager.maxGlobalLongSizes(token);
        amounts[i * propsLength + 10] = vault.getMinPrice(token);
        amounts[i * propsLength + 11] = vault.getMaxPrice(token);
        amounts[i * propsLength + 12] = vault.guaranteedUsd(token);
        amounts[i * propsLength + 13] =
priceFeed.getPrimaryPrice(token, false);
        amounts[i * propsLength + 14] =
priceFeed.getPrimaryPrice(token, true);
    }
    return amounts;
}
```

2. 存在完全一致功能的不同名函数，可能存在部署 gas 费的浪费。

```
function claim() external nonReentrant {
    address account = msg.sender;
    IRewardTracker(feeLpTracker).claimForAccount(account, account);
}
function claimFees() external nonReentrant {
    address account = msg.sender;
    IRewardTracker(feeLpTracker).claimForAccount(account, account);
}
```

3. _setupDecimals 方法可以修改 _decimals，但该方法属性是 internal，并且没有其他方法调用。

```
function _setupDecimals(uint8 decimals_) internal {
    _decimals = decimals_;
}
```

- **安全建议**

1. 仅需要保留一个获取最多数据的函数即可，避免过多的冗余代码浪费部署 gas 费。

2. 删除多余无用代码。

3. 删除无用的代码。

- **修复状态**

1. Rollup.Finance 项目方已确认。
2. Rollup.Finance 项目方已确认。
3. Rollup.Finance 项目方修复。

4.2.21 重入攻击

- **漏洞描述**

攻击者在 Fallback 函数中的外部地址处构建一个包含恶意代码的合约，当合约向此地址发送 代币时，它将调用恶意代码，Solidity 中的 call.value()函数在被用来发送代币时会消耗他接收到的所有 gas，所以当调用 call.value()函数发送代币的操作发生在实际减少发送者账户余额之前时，将会产生重入攻击。由于重入漏洞导致了著名的 The DAO 攻击事件。

合约中 `_transferOutETH` 方法去执行 `_receiver.sendValue(_amountOut);`，这里的 `_receiver` 为用户传进来的地址，可以执行其他逻辑或者回调， `_receiver` 如果是合约地址，存在重入风险， 暂未未发现具体利用点。 多个方法会调用 `_transferOutETH` 方法进行转账。

```
function _transferOutETH(uint256 _amountOut, address payable _receiver)
private {
    IWETH(weth).withdraw(_amountOut);
    _receiver.sendValue(_amountOut);
}
```

- **安全建议**

目前暂无发现可用重入点，但后续如果要修改合约代码，则需要为每一个调用该函数的外部函数添加防止重入机制。

- **审计结果： 通过**

4.2.2 变量更新问题

- **风险描述**

变量更新问题一般发生在奖励和转账阶段，如果某用户获取了自己应得奖励，但奖励发送后，合约内部并未对奖励的变量进行及时更新，导致奖励金额一直存在，该漏洞如果被攻击者恶意利用，或可导致异常资金流失及市场稳定性动摇。

- **审计结果： 通过**

4.2.4 浮点数和数值精度

- **漏洞描述**

在 Solidity 中不支持浮点型，也不完全支持定长浮点型，除法运算的结果会四舍五舍，如果出现小数，小数点后的部分都会被舍弃，只取整数部分，例如直接用 5 除以 2，结果为 2。如果在代币的运算中出现运算结果小于 1 的情况，比如 4.9 个代币也会被约等于 4 个，带来一定程度上的精度流失。由于代币的经济属性，精度的流失就相当于资产的流失，所以这在交易频繁的代币上会带来积少成多的问题。

- **审计结果： 通过**

4.2.3 整数溢出

- **风险描述**

Same Finance 合约，getArgID 方法中输入地址后，会对该地址对应数值进行减 1 计算，如果该地址在 argID[]数组中为空，那么 argID[addr]为 0，这里进行减 1 运算后，会发生整数下溢的情况。

- **审计结果： 通过**

4.2.5 默认可见性

- **漏洞描述**

在 Solidity 中，合约函数的可见性默认是 public。因此，不指定任何可见性的函数就可以由用户在外部调用。当开发人员错误地忽略应该是私有的功能的可见性说明符时，或者是只能在合约本身内调用的可见性说明符时，将导致严重漏洞。在 Parity 多签名钱包遭受的第一次黑客攻击中就是因为未设置函数的可见性，默认为 public，导致大量资金被盗。

- **审计结果： 通过**

4.2.6 tx.origin 身份验证

- **漏洞描述**

tx.origin 是 Solidity 的一个全局变量，它遍历整个调用栈并返回最初发送调用（或事务）的帐户的地址。在智能合约中使用此变量进行身份验证会使合约容易受到类似网络钓鱼的攻击。

- **审计结果： 通过**

4.2.7 错误的构造函数

- **漏洞描述**

在 solidity 智能合约中的 0.4.22 版本之前，所有的合约和构造函数同名。编写合约时，如果构造函数名和合约名不相同，合约会添加一个默认的构造函数，自己设置的构造函数就会被当做普通函数，导致自己原本的合约设置未按照预期执行，这可能会导致可怕的后果，特别是如果构造函数正在执行有权限的操作。

- **审计结果： 通过**

4.2.8 未检验返回值

- **漏洞描述**

在 Solidity 中存在三种向一个地址发送代币的方法：transfer(), send(), call.value()。他们的区别在于 transfer 函数发送失败时会抛出异常 throw，将交易状态回滚，花费 2300gas；send 函数发送失败时返回 false，花费 2300gas；call.value 方法发送失败时返回 false，调用花费全部 gas，将导致重入攻击风险。如果在合约代码中使用 send 或者 call.value 方法进行代币发送时未检查方法返回值，如果发生错误时，合约会继续执行后面得代码，将导致以为的结果。

- **审计结果： 通过**

4.2.9 不安全的随机数

- **漏洞描述**

区块链上的所有交易都是确定性的状态转换操作，没有不确定性，这最终意味着在区块链生态系统内不存在熵或随机性的来源。所以咋 Solidity 中没有 rand() 这种随机数功能。很多开发者使用未来的块变量，如区块哈希值，时间戳，区块高低或是 Gas 上限等来生成随机数，这些量都是由挖矿的矿工控制的，因此并不是真正随机的，因此使用过去或现在的区块变量产生随机数可能导致破坏性漏洞。

- **审计结果： 通过**

4.2.10 时间戳依赖

- **漏洞描述**

在区块链中，数据块时间戳（block.timestamp）被用于各种应用，例如随机数的函数，锁定一段时间的资金以及时间相关的各种状态变化的条件语句。矿工有能力根据需求调整时间戳，比如 block.timestamp 或者别名 now 可以由矿工操纵。如果在智能合约中使用错误的块时间戳，这可能会导致严重漏洞。如果合约不是特别关心矿工对区块时间戳的操纵，这可能是不必要的，但是在开发合约时应该注意这一点。

- **审计结果： 通过**

4.2.11 交易顺序依赖

- **漏洞描述**

在区块链中，矿工会选择来自该矿池的哪些交易将包含在该区块中，这通常是由 gasPrice 交易决定的，矿工将选择交易费最高的交易打包进区块。由于区块中的交易信息对外公开，攻击者可以观察事务池中是否存在可能包含问题解决方案的事务，修改或撤销攻击者的权限或更改合约中的对攻击者不利的状态。然后，攻击者可以从这个事务中获取数据，并创建一个更高级别的事务 gasPrice 并在原始之前将其交易包含在一个区块中，这样将抢占原始事务解决方案。

- **审计结果： 通过**

4.2.12 Delegatecall 函数调用

- **漏洞描述**

在 Solidity 中，delegatecall 函数时标准消息调用方法，但在目标地址中的代码会在调用合约的环境下运行，也就是说，保持 msg.sender 和 msg.value 不变。该功能支持实现库，开发人员可以为未来的合约创建可重用的代码。库中的代码本身可以是安全的，无漏洞的，但是当在另一个应用的环境中运行时，可能会出现新的漏洞，所以使用 delegatecall 函数时可能会导致意外的代码执行。

- **审计结果： 通过**

4.2.13 Call 函数调用

- **漏洞描述**

Call 函数跟 delegatecall 函数相似，都是智能合约编写语言 Solidity 提供的底层函数，用来与外部合约或者库进行交互，但是用 call 函数方法来处理对合约的外部标准信息调用（Standard Message Call）时，代码在外部合约/功能的环境中运行。此类函数使用时需要对调用参数的安全性进行判定，建议谨慎使用，攻击者可以很容易地借用当前合约的身份来进行其他恶意操作，导致严重漏洞。

- **审计结果： 通过**

4.2.14 拒绝服务

- **漏洞描述**

拒绝服务攻击的原因类别比较广泛，其目的就是让用户在一段时间内或永久地在某些情况下使合约无法正常运行，包括在作为交易接收方时的恶意行为，人为增加计算功能所需 gas 导致 gas 耗尽（比如控制 for 循环中的变量大小），滥用访问控制访问合约的 private 组件，在合约中拥有特权的 owner 被修改，基于外部调用的进展状态，利用混淆和疏忽等都能导致拒绝服务攻击。

- **审计结果： 通过**

4.2.16 假充值漏洞

- **漏洞描述**

在代币交易回条状态是成功还是失败（true or false），取决于交易事务执行过程中是否抛出了异常（比如使用了 require/assert/revert/throw 等机制）。当用户调用代币合约的 transfer 函数进行转账时，如果 transfer 函数正常运行未抛出异常，无聊转账交易是否成功，该交易的回执状态就是成功即 true。那么有些代币合约的 transfer 函数对转账发起人(msg.sender)的余额检查用的是 if 判断方式，当 balances[msg.sender] < _value 时进入 else 逻辑部分并 return false，最终没有抛出异常，但是交易回执是成功的，那么我们认为仅 if/else 这种温和的判断方式在 transfer 这类敏感函数场景中是一种不严谨的编码方式，将导致相关中心化交易所、中心化钱包、代币合约的假充值漏洞。

- **审计结果： 通过**

4.2.17 短地址攻击漏洞

- **漏洞描述**

在 Solidity 智能合约中，将参数传递给智能合约时，参数将根据 ABI 规范进行编码。EVM 运行攻击者发送比预期参数长度短的编码参数。例如在交易所或者钱包转账时，需要发送转账地址 address 和转账金额 value，攻击者可以发送 19 字节的地址而不是标准的 20 字节地址，在这种情况下，EVM 会将 0 填到编码参数的末尾以补成预期的长度，这将导致最后转账金额参数 value 的溢出，从而改变原本转账金额。

- **审计结果： 通过**

4.2.18 未初始化的存储指针

- **漏洞描述**

EVM 既用 storage 来存储变量，也用 memory 来存储变量，函数内的局部变量根据它们的类型默认用 storage 或 memory 存储，在 Solidity 的工作方式里面，状态变量按它们出现在合约中的顺序存储在合约的 Slot 中 ，未初始化的局部 storage 变量可能会指向合约中的其他意外存储变量，从而导致有意或无意的漏洞。

- **审计结果： 通过**

4.2.19 冻结账户绕过

- **漏洞描述**

在合约中的转账操作代码中，检测合约代码中是否存在对转账账户冻结状态检查的逻辑功能，如果转账账户已经冻结，是否可被绕过的风险。

- **审计结果： 通过**

4.2.20 未初始化

- **漏洞描述**

在合约中的 initialize 函数可被其他攻击者抢在 owner 之前调用，从而初始化管理员地址。

- **审计结果： 通过**

5.安全审计工具

工具名称	功能
Oyente	可以用来检测智能合约中常见 bug
securify	可以验证以太坊智能合约的常见类型
MAIAN	可以查找多个智能合约漏洞并进行分类
零时内部工具包	零时(鹰眼系统)自研发工具包+https://audit.noneage.com

免责声明：

零时科技仅就本报告出具之前发生或存在的事实出具报告并承担相应责任，对于出具报告之后发生的事实由于无法判断智能合约安全状态，因此不对此承担责任。零时科技对该项目约定内的安全审计项进行安全审计，不对该项目背景及其他情况进行负责，项目方后续的链上部署以及运营方式不在本次审计范围。本报告只基于信息提供者截止出具报告时向零时科技提供的信息进行安全审计，对于此项目的信息有隐瞒，或反映的情况与实际情况不符的，零时科技对由此而导致的损失和不利影响不承担任何责任。

市场有风险，投资需谨慎，此报告仅对智能合约代码进行安全审计和结果公示，不作投资建议和依据。

零时科技
NONEAGE

邮　箱：support@noneage.com
官　网：www.noneage.com
微　博：weibo.com/noneage