智能合约安全审计报告

Xensa 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 ：2021-10-18

Xensa 智能合约安全审计报告

1.概述

零时科技安全团队于 2021 年 09 月 22 日，接到 **Xensa 项目**的安全审计需求，团队于 2021 年 10 月 18 日对 **Xensa 智能合约**审计完成，审计过程中零时科技安全审计专家与 Xensa 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 Xensa 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 Xensa 智能合约安全审计结果：通过安全审计。

合约报告 MD5：994C5C461584CEE1CC1DDD9F87D80C34

2.项目背景

2.1 项目简介

项目名称：Xensa

合约类型：DeFi

代码语言：Solidity

源码地址：https://github.com/XensaFi/xensa-protocol

审计版本：commit 96eec8b17e613512194b1e22d1154801e1b67f1d

合约文件：Migrations.sol，TokenDistributor.sol，FeeProvider.sol，stake.sol，XToken.sol，XensaAddressesProvider.sol，UintStorage.sol，AddressStorage.sol，OKToracle.sol，XensaParametersProvider.sol，XensaManager.sol，GenericOracleI.sol，InterestRateOracle.sol，PriceOracle.sol，MockAggregatorUSDC.sol，MockAggregatorKNC.sol，MockAggregatorUSDT.sol，MockAggregatorMKR.sol，MockAggregatorWBTC.sol，MockAggregatorDAI.sol，MockAggregatorLINK.sol，MockAggregatorBase.sol，MockAggregatorSUSD.sol，MockAggregatorREP.sol，MockAggregatorZRX.sol，MockAggregatorMANA.sol，MockAggregatorTUSD.sol，MockAggregatorBAT.sol，MockXensaCore.sol，MockFlashLoanReceiver.sol，XensaMinter.sol，XensaQuery.sol，XensaToken.sol，Xensa.sol，XensaDataProvider.sol，XensaCore.sol，XensaConfigurator.sol，XensaLiquidationManager.sol，IERC20DetailedBytes.sol，ChainlinkProxyPriceProvider.sol，

WalletBalanceProvider.sol，Initializable.sol，IFlashLoanReceiver.sol，
FlashLoanReceiverBase.sol，DefaultReserveInterestRateStrategy.sol

2.2 审计范围

Xensa 官方提供合约文件及文件对应 MD5：

Migrations.sol	6F3DC9936EC2CFF1666C76993E7D6531
TokenDistributor.sol	E732C65EF1BD52598DA4A9FD2B60ECB8
FeeProvider.sol	49246E27B8E47158A8CDCA62739AC92D
stake.sol	B214A988B4AEAE786C4E551FA1EBE840
XToken.sol	AD64215F47AE49C6ECDC8AE5ACD17727
XensaAddressesProvider.sol	C37B87E93D240DFEF2FCCFDD58202772
UintStorage.sol	EFC20BC1D7808ABBD115B8BAFA1323AD
AddressStorage.sol	C12425238ADD3708B841B4B5F278268E
XensaParametersProvider.sol	1BFDF7ACB6C879FA5C14D075614FFBC4
XensaManager.sol	97D6D92645658519E31E7A83D826C6F0
GenericOracleI.sol	79ECF7E540C043149B5B5DDEF77296BF
InterestRateOracle.sol	1D0CF378C236D9F5CD8A49567118D1DF
PriceOracle.sol	EEC7B9BA2799CF07BBCC9BABB92632BC
MockAggregatorUSDC.sol	9E8ADD4DCC5D77057BDD467617DE0D11
MockAggregatorKNC.sol	C16CC947EDD407A9776D65D5E3D18C9A
MockAggregatorUSDT.sol	7F8B8DE0066750390CCEDF03FF75F693
MockAggregatorMKR.sol	4C3A3F24AFE8BA6ABED7C18AC8118DCD
MockAggregatorWBTC.sol	1E573ED3C68AE9AC413151AF1E796EB2
MockAggregatorDAI.sol	F81E4FCB2DCDD6FB7BD7F4A5E022A410
MockAggregatorLINK.sol	3C4334979EE33384C4373B374562340E
MockAggregatorBase.sol	B4C5C9A201AC94E7F70C2ED5A2351F5E

MockAggregatorSUSD.sol	E507654C3AED98F00A1FA649D2555888
MockAggregatorREP.sol	0A449B0242E8977274D99A6242CB9FC6
MockAggregatorZRX.sol	E74A7E97D4A0B4AAEB14C031E950426F
MockAggregatorMANA.sol	94AF38D49E41B460779A4D67CD38CA9E
MockAggregatorTUSD.sol	CE030CD159454646FD4A3DED992B9A4B
MockAggregatorBAT.sol	5C63992FDB80D882C52BB0903739238D
MockXensaCore.sol	2DB82ADD1F347E84AE89F1A5EFD20408
MockFlashLoanReceiver.sol	4D744D27E6F5D7022DB6655A09743F97
XensaMinter.sol	444A4FD2259A1F95E813C17B0FA1CF3C
XensaQuery.sol	CB132544704137B85BD6148D5F00D280
XensaToken.sol	B6C864BC87FA717E886339A7C19E61B0
Xensa.sol	CB22D37B05FDA19B704386FFC594F5F0
XensaDataProvider.sol	882D88EA0A33E0D52437F3CF0A5C01A9
XensaCore.sol	6C7ECD16CE1AC06F10F2878FF8999F87
XensaConfigurator.sol	2DB72A118CCEE33954CC4B49836618A8
XensaLiquidationManager.sol	6B260A5B824B6CA5A031C78FBDA6A42A
OKToracle.sol	563D3AA5039DFE0263B026477808F35D
IERC20DetailedBytes.sol	602BA30C825913860BAA77B509877719
ChainlinkProxyPriceProvider.sol	4BB9C5F3F4D4BA35C54F4C9690BE67B5
WalletBalanceProvider.sol	91239CDE8B1F794E19FF897EAF74C83B
Initializable.sol	5125A44096787D8888A854A26DCE79B8
IFlashLoanReceiver.sol	0FFB754266F5BC02DFBC8B7A7EBBED1A
FlashLoanReceiverBase.sol	99ED887132A82057889FC6E93778C042
DefaultReserveInterestRateStrategy.sol ->D142CC9499D421DCAFD244D738FCB1B3	

3.合约架构分析

3.1 目录结构

```
└─Xensa contracts
    │  Migrations.sol
    ├──configuration
    │      AddressStorage.sol
    │      UintStorage.sol
    │      XensaAddressesProvider.sol
    │      XensaManager.sol
    │      XensaParametersProvider.sol
    ├──fees
    │      FeeProvider.sol
    │      stake.sol
    │      TokenDistributor.sol
    ├──flashloan
    │   ├──base
    │   │      FlashLoanReceiverBase.sol
    │   └──interfaces
    │          IFlashLoanReceiver.sol
    ├──mint
    │      XensaMinter.sol
    │      XensaQuery.sol
    │      XensaToken.sol
    ├──misc
    │      ChainlinkProxyPriceProvider.sol
    │      IERC20DetailedBytes.sol
```

```
|   OKToracle.sol
|   WalletBalanceProvider.sol
├──mocks
|   ├──flashloan
|   |   MockFlashLoanReceiver.sol
|   ├──oracle
|   |   |   GenericOracleI.sol
|   |   |   InterestRateOracle.sol
|   |   |   PriceOracle.sol
|   |   └──CLAggregators
|   |       MockAggregatorBase.sol
|   |       MockAggregatorBAT.sol
|   |       MockAggregatorDAI.sol
|   |       MockAggregatorKNC.sol
|   |       MockAggregatorLINK.sol
|   |       MockAggregatorMANA.sol
|   |       MockAggregatorMKR.sol
|   |       MockAggregatorREP.sol
|   |       MockAggregatorSUSD.sol
|   |       MockAggregatorTUSD.sol
|   |       MockAggregatorUSDC.sol
|   |       MockAggregatorUSDT.sol
|   |       MockAggregatorWBTC.sol
|   |       MockAggregatorZRX.sol
|   └──upgradeability
|       MockXensaCore.sol
```

```
├──tokenization
│       XToken.sol
└──xensa
│       DefaultReserveInterestRateStrategy.sol
│       Xensa.sol
│       XensaConfigurator.sol
│       XensaCore.sol
│       XensaDataProvider.sol
│       XensaLiquidationManager.sol
```

3.2 合约详情

XensaAddressesProvider Contract

方法名称	方法传参	方法属性
getXensa	none	public
setXensaImpl	address _xensa	onlyOwner
getXensaCore	none	public
setXensaCoreImpl	address _xensaCore	onlyOwner
getXensaConfigurator	none	public
setXensaConfiguratorImpl	address _configurator	onlyOwner
getXensaDataProvider	none	public
setXensaDataProviderImpl	address _provider	onlyOwner
getXensaParametersProvider	none	public
setXensaParametersProviderImpl	address _parametersProvider	onlyOwner
getFeeProvider	none	public
setFeeProviderImpl	address _feeProvider	onlyOwner
getXensaLiquidationManager	none	public
setXensaLiquidationManager	address _manager	onlyOwner
getXensaManager	none	public
setXensaManager	address _xensaManager	onlyOwner
getPriceOracle	none	public
setPriceOracle	address _priceOracle	onlyOwner
getInterestRateOracle	none	public
setInterestRateOracle	address _interestRateOracle	onlyOwner
getTokenDistributor	none	public
setTokenDistributor	address _tokenDistributor	onlyOwner
getXensaMinter	none	public
setXensaMinter	address _xensaMinter	onlyOwner
updateImplInternal	bytes32 _id address _newAddress	internal

XensaManager Contract

方法名称	方法传参	方法属性
initAddressProvider	XensaAddressesProvider _addressesProvider	onlyOwner
getRevision	none	internal
initReserve	address _reserve uint8 _underlyingAssetDecimals address _interestRateStrategyAddress	onlyOwner
initReserveWithData	address _reserve string _xTokenName string _xTokenSymbol uint8 _underlyingAssetDecimals address _interestRateStrategyAddress	onlyOwner
removeLastAddedReserve	address _reserveToRemove	onlyOwner
enableBorrowingOnReserve	address _reserve bool _stableBorrowRateEnabled	onlyOwner
disableBorrowingOnReserve	address _reserve	onlyOwner
enableReserveAsCollateral	address _reserve uint256 _baseLTVasCollateral uint256 _liquidationThreshold uint256 _liquidationBonus	onlyOwner
disableReserveAsCollateral	address _reserve	onlyOwner
enableReserveStableBorrowRate	address _reserve	onlyOwner
disableReserveStableBorrowRate	address _reserve	onlyOwner
activateReserve	address _reserve	onlyOwner
deactivateReserve	address _reserve	onlyOwner
freezeReserve	address _reserve	onlyOwner
unfreezeReserve	address _reserve	onlyOwner
setReserveBaseLTVasCollateral	address _reserve uint256 _ltv	onlyOwner
setReserveLiquidationThreshold	address _reserve uint256 _threshold	onlyOwner

方法名称	方法传参	方法属性
setReserveLiquidationBonus	address _reserve uint256 _bonus	onlyOwner
setReserveDecimals	address _reserve uint256 _decimals	onlyOwner
setReserveInterestRateStrategyAddress	address _reserve address _rateStrategyAddress	onlyOwner
refreshXensaCoreConfiguration	none	onlyOwner

UintStorage Contract

方法名称	方法传参	方法属性
getUint	bytes32 _key	public
_setUint	bytes32 _key uint256 _value	internal

AddressStorage Contract

方法名称	方法传参	方法属性
getAddress	bytes32 _key	public
_setAddress	bytes32 _key address _value	internal

TokenDistributor Contract

方法名称	方法传参	方法属性
getDistribution	none	public
getRevision	none	internal
internalTrade	address _from uint256 _amount	internal
internalBurn	uint256 _amount	internal

XensaStaking Contract

方法名称	方法传参	方法属性
owner	none	public
isOwner	none	public
_renounceOwnership	none	internal
checkContract	address _account	internal
totalSupply	none	external
balanceOf	address account	external
transfer	address recipient uint256 amount	external
allowance	address owner address spender	external
increaseAllowance	address spender uint256 addedValue	external
decreaseAllowance	address spender uint256 subtractedValue	external
approve	address spender uint256 amount	external
transferFrom	address sender address recipient uint256 amount	external
name	none	external
symbol	none	external
decimals	none	external
permit	address owner address spender uint256 amount uint256 deadline uint8 v bytes32 r bytes32 s	external
nonces	address owner	external
version	none	external
permitTypeHash	none	external
domainSeparator	none	external
_min	uint _a uint _b	internal
_max	uint _a uint _b	internal
decMul	uint x uint y	internal
_decPow	uint _base uint _minutes	internal
_getAbsoluteDifference	uint _a uint _b	internal
_computeNominalCR	uint _coll uint _debt	internal

方法名称	方法传参	方法属性
_computeCR	uint _coll uint _debt uint _price	internal
addAsset	address _assetAddress	onlyOwner
stake	uint _XensaAmount	external
unstake	uint _XensaAmount	external
assetIncome	address _assetAddress	public
updateAssetsFeePerStake	none	internal
_updateAssetFeePerStake	address asset	internal
_computeAssetFeePerStake	address asset	public
getPendingGain	address _user address asset	external
_getPendingGain	address _user address asset	internal
_updateUserSnapshots	none	internal
_sendETHGainToUser	uint ETHGain	internal
_requireCallerIsTroveManager	none	internal
_requireCallerIsBorrowerOperations	none	internal
_requireCallerIsActivePool	none	internal
_requireUserHasStake	uint currentStake	internal
_requireNonZeroAmount	uint _amount	internal

Migrations Contract

方法名称	方法传参	方法属性
setCompleted	uint completed	public

XensaQuery Contract

方法名称	方法传参	方法属性
getGroupInfo	uint256 _pid uint256 _gid	external
pendingXensa	uint256 _pid uint256 _gid address _user	external
earnings	uint256 pid uint256 gid uint256 duringBlocks uint256 baseReservsePrice uint256 mintReservePrice	public
mintTokenPerBlock	uint256 pid uint256 gid	public
mintedToken	uint256 pid uint256 gid uint256 poolStart uint256 poolEnd	public

XensaMiner Contract

方法名称	方法传参	方法属性
setLp	address addressLP	onlyOwner
mint	address _to uint256 _amount	internal
poolLength	none	public
createPool	uint256 _pid uint256 _poolCap uint256 _startBlock uint256 _endBlock	onlyOwner
setGroup	uint256 _pid uint256 _gid uint256 _allocPoint	onlyOwner
getMultiplier	uint256 _from uint256 _to uint256 _start uint256 _end	internal
massUpdateGroups	uint256 _pid	internal
calculateMGR	uint256 point uint256 base	internal
updateUserProtectDuration	address userAddr UserInfo u uint256 amount uint256 pending uint256 poolPrice bool unlockedOnly	internal
updateGroup	uint256 _pid uint256 _gid	internal
getPoolInfo	uint256 _pid	public
getGroupInfo	uint256 _pid uint256 _gid	public
_deposit	uint256 _pid uint256 _gid address u uint256 _amount	internal
_withdraw	uint256 _pid uint256 _gid address u uint256 _amount bool unlockedOnly	internal
pendingXensa	uint256 _pid uint256 _gid address _user	public
selectPool	none	internal
getUserAmount	uint256 _gid address _user	public
_priceRate	PoolInfo p Group g UserInfo u	internal
userGainPrice	uint256 _pid uint256 _gid address _u uint256 _amount bool isDeposit	internal
updatePricePerStake	uint256 _pid uint256 amount	internal
updateGroupStake	uint256 _pid uint256 _gid uint256 amount	internal
mintXensaToken	address _reserve address _user uint256 _gid uint256 _amount	external

方法名称	方法传参	方法属性
	uint256 _reserveDEC uint256 _price	
getUserReserveAVP	uint256 _pid uint256 _gid address _reserve address _user	public
_mintXensaToken	address _user uint256 _pid uint256 _gid uint256 _amount	internal
withdrawXensaToken	address _reserve address _user uint256 _gid uint256 _amount uint256 _dec bool unlockedOnly	external
_withdrawXensaToken	address _user uint256 _pid uint256 _gid uint256 _amount bool unlockedOnly	internal
_withdrawPendingXensaToken	uint256 _gid bool unlockedOnly	public
withdrawPendingXensaToken	bool unlockedOnly	public
setPoolInited	uint256 _pid	onlyOwner
deposit	uint256 _pid uint256 _gid address u uint256 _amount	onlyOwner
withdraw	uint256 _pid uint256 _gid address u uint256 _amount	onlyOwner

XensaToken Contract

方法名称	方法传参	方法属性
XensaTokenMint	address _to uint256 _amount	internal
totalSupply	none	public
cap	none	public

ChainlinkProxyPriceProvider Contract

方法名称	方法传参	方法属性
setFallbackOracle	address _fallbackOracle	onlyOwner
internalSetFallbackOracle	address _fallbackOracle	internal
getAssetPrice	address _asset	public
getSourceOfAsset	address _asset	external
getFallbackOracle	none	external

OKOracle Contract

方法名称	方法传参	方法属性
registerRequesterViewer	none	external
latestRoundData	string priceType address dataSource	external
get	string priceType address source	external
getOffchain	string priceType address source	external
getCumulativePrice	string priceType address source	external
changeSourceRecipient	address _recipient	external
changeFeederRecipient	address _recipient	external
postMining	address requester bytes message bytes signature	external
transferCredit	uint256 amount address to	external
symbol	none	external
getAssetPrice	address _asset	external
internalSetFallbackOracle	address _fallbackOracle	internal
setAssetSource	address _asset string _symbol	onlyOwner
unsetAssetSource	address _asset	onlyOwner
internalSetAssetsSource	address _asset string _symbol	internal
internalUnSetAssetsSource	address _asset	internal
getLatestPrice	string priceType	public
getBasePrice	none	public
getAssetPrice	address _asset	external

WalletBalanceProvider Contract

方法名称	方法传参	方法属性
balanceOf	address _user address _token	public
getUserWalletBalances	address _user	public

IFlashLoanReceiver Contract

方法名称	方法传参	方法属性
executeOperation	address _reserve uint256 _amount uint256 _fee bytes _params	external

XensaConfigurator Contract

方法名称	方法传参	方法属性
getRevision	none	internal
initialize	XensaAddressesProvider _addressesProvider	public
initReserve	address _reserve uint8 _underlyingAssetDecimals address _interestRateStrategyAddress	onlyXensaManager
initReserveWithData	address _reserve string _xTokenName string _xTokenSymbol uint8 _underlyingAssetDecimals address _interestRateStrategyAddress	onlyXensaManager
removeLastAddedReserve	address _reserveToRemove	onlyXensaManager
enableBorrowingOnReserve	address _reserve bool _stableBorrowRateEnabled	onlyXensaManager
disableBorrowingOnReserve	address _reserve	onlyXensaManager
enableReserveAsCollateral	address _reserve uint256 _baseLTVasCollateral uint256 _liquidationThreshold uint256 _liquidationBonus	onlyXensaManager
disableReserveAsCollateral	address _reserve	onlyXensaManager
enableReserveStableBorrowRate	address _reserve	onlyXensaManager
disableReserveStableBorrowRate	address _reserve	onlyXensaManager
activateReserve	address _reserve	onlyXensaManager
deactivateReserve	address _reserve	onlyXensaManager

方法名称	方法传参	方法属性
freezeReserve	address _reserve	onlyXensaManager
unfreezeReserve	address _reserve	onlyXensaManager
setReserveBaseLTVasCollateral	address _reserve uint256 _ltv	onlyXensaManager
setReserveLiquidationThreshold	address _reserve uint256 _threshold	onlyXensaManager
setReserveLiquidationBonus	address _reserve uint256 _bonus	onlyXensaManager
setReserveDecimals	address _reserve uint256 _decimals	onlyXensaManager
setReserveInterestRateStrategyAddress	address _reserve address _rateStrategyAddress	onlyXensaManager
refreshXensaCoreConfiguration	none	onlyXensaManager

DefaultReserveInterestRateStrategy Contract

方法名称	方法传参	方法属性
getBaseVariableBorrowRate	none	external
getVariableRateSlope1	none	external
getVariableRateSlope2	none	external
getStableRateSlope1	none	external
getStableRateSlope2	none	external
calculateInterestRates	address _reserve uint256 _availableLiquidity uint256 _totalBorrowsStable uint256 _totalBorrowsVariable uint256 _averageStableBorrowRate	external
getOverallBorrowRateInternal	uint256 _totalBorrowsStable uint256 _totalBorrowsVariable uint256 _currentVariableBorrowRate uint256 _currentAverageStableBorrowRate	internal

XToken Contract

方法名称	方法传参	方法属性
_transfer	address _from address _to uint256 _amount	internal
redirectInterestStream	address _to	external
redirectInterestStreamOf	address _from address _to	external
allowInterestRedirectionTo	address _to	external
redeem	uint256 _amount	external
mintOnDeposit	address _account uint256 _amount	external
burnOnLiquidation	address _account uint256 _value	external
transferOnLiquidation	address _from address _to uint256 _value	external
balanceOf	address _user	public
principalBalanceOf	address _user	external
totalSupply	none	public
isTransferAllowed	address _user uint256 _amount	public
getUserIndex	address _user	external
getInterestRedirectionAddress	address _user	external
getRedirectedBalance	address _user	external
cumulateBalanceInternal	address _user	internal

方法名称	方法传参	方法属性
updateRedirectedBalanceOfRedirectionAddressInternal	address _user uint256 _balanceToAdd uint256 _balanceToRemove	internal
calculateCumulatedBalanceInternal	address _user uint256 _balance	internal
executeTransferInternal	address _from address _to uint256 _value	internal
redirectInterestStreamInternal	address _from address _to	internal
resetDataOnZeroBalanceInternal	address _user	internal

XensaLiquidationManager Contract

方法名称	方法传参	方法属性
init	XensaAddressesProvider _addressesProvider	onlyOwner
getRevision	none	internal
liquidationCall	address _collateral address _reserve address _user address _caller uint256 _purchaseAmount bool _receiveXToken	external
calculateAvailableCollateralToLiquidate	address _collateral address _principal uint256 _purchaseAmount uint256 _userCollateralBalance	internal

Xensa Contract

方法名称	方法传参	方法属性
getRevision	none	internal
initialize	XensaAddressesProvider _addressesProvider	public
deposit	address _reserve uint256 _amount uint16 _referralCode	onlyAmountGreaterThanZero
redeemUnderlying	address _reserve address _user uint256 _amount uint256 _xTokenBalanceAfterRedeem	onlyAmountGreaterThanZero
borrow	address _reserve uint256 _amount uint256 _interestRateMode uint16 _referralCode	onlyAmountGreaterThanZero
repay	address _reserve uint256 _amount address _onBehalfOf	onlyAmountGreaterThanZero
swapBorrowRateMode	address _reserve	onlyUnfreezedReserve
rebalanceStableBorrowRate	address _reserve address _user	onlyActiveReserve
setUserUseReserveAsCollateral	address _reserve bool _useAsCollateral	onlyUnfreezedReserve
liquidationCall	address _collateral address _reserve address _user address _caller uint256 _purchaseAmount bool _receiveXToken	onlyActiveReserve
flashLoan	address _receiver address _reserve uint256 _amount bytes _params	onlyAmountGreaterThanZero
getReserveConfigurationData	address _reserve	external
getReserveData	address _reserve	external
getUserAccountData	address _user	external

方法名称	方法传参	方法属性
getUserReserveData	address _reserve address _user	external
getReserves	none	external
requireReserveActiveInternal	address _reserve	internal
requireReserveNotFreezedInternal	address _reserve	internal
requireAmountGreaterThanZeroInternal	uint256 _amount	internal

XensaParametersProvider Contract

方法名称	方法传参	方法属性
getRevision	none	internal
initialize	address _addressesProvider	public
getMaxStableRateBorrowSizePercent	none	external
getRebalanceDownRateDelta	none	external
getFlashLoanFeesInBips	none	external

FeeProvider Contract

方法名称	方法传参	方法属性
getRevision	none	internal
initialize	address _addressesProvider	public
calculateLoanOriginationFee	address _user uint256 _amount	external
getLoanOriginationFeePercentage	none	external

FlashLoanReceiverBase Contract

方法名称	方法传参	方法属性
transferFundsBackToPoolInternal	address _reserve uint256 _amount	internal
transferInternal	address _destination address _reserve uint256 _amount	internal
getBalanceInternal	address _target address _reserve	internal

XensaDataProvider Contract

方法名称	方法传参	方法属性
getRevision	none	internal
initialize	XensaAddressesProvider _addressesProvider	public
calculateUserGlobalData	address _user	public
balanceDecreaseAllowed	address _reserve address _user uint256 _amount	external
getHealthFactorAfterDecrease	address _reserve address _user uint256 _amount bool _isDecrease	external
calculateCollateralNeededInETH	address _reserve uint256 _amount uint256 _fee uint256 _userCurrentBorrowBalanceTH uint256 _userCurrentFeesETH uint256 _userCurrentLtv	external
calculateAvailableBorrowsETHInternal	uint256 collateralBalanceETH uint256 borrowBalanceETH uint256 totalFeesETH uint256 ltv	internal
calculateHealthFactorFromBalancesInternal	uint256 collateralBalanceETH uint256 borrowBalanceETH uint256 totalFeesETH uint256 liquidationThreshold	internal
getHealthFactorLiquidationThreshold	none	public
getReserveConfigurationData	address _reserve	external
getReserveData	address _reserve	external
getUserAccountData	address _user	external
getUserReserveData	address _reserve address _user	external
getReserveValues	address _reserve	external

XensaCore Contract

方法名称	方法传参	方法属性
getRevision	none	internal
initialize	XensaAddressesProvider _addressesProvider	public
updateStateOnDeposit	address _reserve address _user uint256 _amount bool _isFirstDeposit	onlyXensa
updateStateOnRedeem	address _reserve address _user uint256 _amountRedeemed bool _userRedeemedEverything	onlyXensa
updateStateOnFlashLoan	address _reserve uint256 _availableLiquidityBefore uint256 _income uint256 _protocolFee	onlyXensa
updateStateOnBorrow	address _reserve address _user uint256 _amountBorrowed uint256 _borrowFee CoreLibrary.InterestRateMode _rateMode	onlyXensa
updateStateOnRepay	address _reserve address _user uint256 _paybackAmountMinusFees uint256 _originationFeeRepaid uint256 _balanceIncrease bool _repaidWholeLoan	onlyXensa
updateStateOnSwapRate	address _reserve address _user uint256 _principalBorrowBalance uint256 _compoundedBorrowBalance uint256 _balanceIncrease CoreLibrary.InterestR	onlyXensa

方法名称	方法传参	方法属性
	ateMode _currentRateMode	
updateStateOnLiquidation	address _principalReserve address _collateralReserve address _user uint256 _amountToLiquidate uint256 _collateralToLiquidate uint256 _feeLiquidated uint256 _liquidatedCollateralForFee uint256 _balanceIncrease bool _liquidatorReceivesXToken	onlyXensaStaff
updateStateOnRebalance	address _reserve address _user uint256 _balanceIncrease	onlyXensa
setUserUseReserveAsCollateral	address _reserve address _user bool _useAsCollateral	onlyXensa
transferToUser	address _reserve address _user uint256 _amount	onlyXensaStaff
transferToFeeCollectionAddress	address _token address _user uint256 _amount address _destination	onlyXensa
liquidateFee	address _token uint256 _amount address _destination	onlyXensaStaff
transferToReserve	address _reserve address _user uint256 _amount	onlyXensaStaff
getUserBasicReserveData	address _reserve address _user	external
isUserAllowedToBorrowAtStable	address _reserve address _user uint256 _amount	external

方法名称	方法传参	方法属性
getUserUnderlyingAssetBalance	address _reserve address _user	public
getReserveInterestRateStrategyAddress	address _reserve	public
getReserveUnit	address _reserve	public
getReserveXTokenAddress	address _reserve	public
getReserveAvailableLiquidity	address _reserve	public
getReserveTotalLiquidity	address _reserve	public
getReserveNormalizedIncome	address _reserve	external
getReserveTotalBorrows	address _reserve	public
getReserveTotalBorrowsStable	address _reserve	external
getReserveTotalBorrowsVariable	address _reserve	external
getReserveLiquidationThreshold	address _reserve	external
getReserveLiquidationBonus	address _reserve	external
getReserveCurrentVariableBorrowRate	address _reserve	external
getReserveCurrentStableBorrowRate	address _reserve	public
getReserveCurrentAverageStableBorrowRate	address _reserve	external
getReserveCurrentLiquidityRate	address _reserve	external
getReserveLiquidityCumulativeIndex	address _reserve	external
getReserveVariableBorrowsCumulativeIndex	address _reserve	external
getReserveConfiguration	address _reserve	external
getReserveDecimals	address _reserve	external
isReserveBorrowingEnabled	address _reserve	external
isReserveUsageAsCollateralEnabled	address _reserve	external
getReserveIsStableBorrowRateEnabled	address _reserve	external
getReserveIsActive	address _reserve	external
getReserveIsFreezed	address _reserve	external
getReserveLastUpdate	address _reserve	external
getReserveUtilizationRate	address _reserve	public
getReserves	none	external
isUserUseReserveAsCollateralEnabled	address _reserve address _user	external

方法名称	方法传参	方法属性
getUserOriginationFee	address _reserve address _user	external
getUserCurrentBorrowRateMode	address _reserve address _user	public
getUserCurrentBorrowRate	address _reserve address _user	internal
getUserCurrentStableBorrowRate	address _reserve address _user	external
getUserBorrowBalances	address _reserve address _user	public
getUserVariableBorrowCumulativeIndex	address _reserve address _user	external
getUserLastUpdate	address _reserve address _user	external
refreshConfiguration	none	onlyXensaConfigurator
initReserve	address _reserve address _xTokenAddress uint256 _decimals address _interestRateStrategyAddress	onlyXensaConfigurator
removeLastAddedReserve	address _reserveToRemove	onlyXensaConfigurator
setReserveInterestRateStrategyAddress	address _reserve address _rateStrategyAddress	onlyXensaConfigurator
enableBorrowingOnReserve	address _reserve bool _stableBorrowRateEnabled	onlyXensaConfigurator
disableBorrowingOnReserve	address _reserve	onlyXensaConfigurator
enableReserveAsCollateral	address _reserve uint256 _baseLTVasCollateral uint256 _liquidationThreshold uint256 _liquidationBonus	onlyXensaConfigurator

方法名称	方法传参	方法属性
disableReserveAsCollateral	address _reserve	onlyXensaConfigurator
enableReserveStableBorrowRate	address _reserve	onlyXensaConfigurator
disableReserveStableBorrowRate	address _reserve	onlyXensaConfigurator
activateReserve	address _reserve	onlyXensaConfigurator
deactivateReserve	address _reserve	onlyXensaConfigurator
freezeReserve	address _reserve	onlyXensaConfigurator
unfreezeReserve	address _reserve	onlyXensaConfigurator
setReserveBaseLTVasCollateral	address _reserve uint256 _ltv	onlyXensaConfigurator
setReserveLiquidationThreshold	address _reserve uint256 _threshold	onlyXensaConfigurator
setReserveLiquidationBonus	address _reserve uint256 _bonus	onlyXensaConfigurator
setReserveDecimals	address _reserve uint256 _decimals	onlyXensaConfigurator
updateReserveStateOnBorrowInternal	address _reserve address _user uint256 _principalBorrowBalance uint256 _balanceIncrease uint256 _amountBorrowed CoreLibrary.InterestRateMode _rateMode	internal
updateUserStateOnBorrowInternal	address _reserve address _user uint256 _amountBorrowed uint256 _balanceIncrease uint256 _fee CoreLibrary.InterestRateMode _rateMode	internal
updateReserveStateOnRepayInternal	address _reserve address _user uint256	internal

方法名称	方法传参	方法属性
	_paybackAmountMinusFees uint256 _balanceIncrease	
updateUserStateOnRepayInternal	address _reserve address _user uint256 _paybackAmountMinusFees uint256 _originationFeeRepaid uint256 _balanceIncrease bool _repaidWholeLoan	internal
updateReserveStateOnSwapRateInternal	address _reserve address _user uint256 _principalBorrowBalance uint256 _compoundedBorrowBalance CoreLibrary.InterestRateMode _currentRateMode	internal
updateUserStateOnSwapRateInternal	address _reserve address _user uint256 _balanceIncrease CoreLibrary.InterestRateMode _currentRateMode	internal
updatePrincipalReserveStateOnLiquidationInternal	address _principalReserve address _user uint256 _amountToLiquidate uint256 _balanceIncrease	internal
updateCollateralReserveStateOnLiquidationInternal	address _collateralReserve	internal
updateUserStateOnLiquidationInternal	address _reserve address _user uint256 _amountToLiquidate uint256 _feeLiquidated uint256 _balanceIncrease	internal

方法名称	方法传参	方法属性
updateReserveStateOnRebalanceInternal	address _reserve address _user uint256 _balanceIncrease	internal
updateUserStateOnRebalanceInternal	address _reserve address _user uint256 _balanceIncrease	internal
updateReserveTotalBorrowsByRateModeInternal	address _reserve address _user uint256 _principalBalance uint256 _balanceIncrease uint256 _amountBorrowed CoreLibrary.InterestRateMode _newBorrowRateMode	internal
updateReserveInterestRatesAndTimestampInternal	address _reserve uint256 _liquidityAdded uint256 _liquidityTaken	internal
transferFlashLoanProtocolFeeInternal	address _token uint256 _amount	internal
refreshConfigInternal	none	internal
addReserveToListInternal	address _reserve	internal

4.审计详情

4.1 风险分布

风险名称	风险级别	修复状态
未添加事件	无	正常
嵌套映射问题	无	正常
错误的对象初始化	无	正常
未添加错误提示	无	正常
任意用户绕过判断	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
逻辑设计缺陷	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
重入攻击	无	正常

4.2 风险审计详情

4.2.1 未添加事件

- **风险描述**

如果合约内多个方法存在敏感操作，但未添加事件记录，管理员及用户操作后，无法确认操作内容及事件追溯，不能及时了解方法内部调用详情，可能出现用户不信任问题。

- **审计结果： 通过**

4.2.2 嵌套映射问题

- **风险描述**

嵌套映射取值一般为键值对，某些合约在嵌套映射取值时，对键输入参数混淆，导致取值错误或者异常，从而导致合约不能正常运行。

- **审计结果： 通过**

4.2.3 错误的对象初始化

- **风险描述**

合约中变量对象没有对其状态发生改变,而使用 storage 初始化变量会大大增加 gas 消耗量，该操作有超过上限回退的风险，且会增加内存被覆写的风险。

- **审计结果： 通过**

4.2.4 未添加错误提示

- **风险描述**

如果合约内 require 条件判断未添加错误提示，管理员及用户操作后，无法确认操作错误的原因，不能及时对操作进行准确修改，可能出现代码实用性不高。

- **审计结果： 通过**

4.2.5 任意用户绕过判断

- **风险描述**

合约存取或转账方法为任意用户调用时，当传入参数可控，应避免使用该参数进行独立判断，或者添加实际条件以避免用户参数可控造成的任意绕过判断。

- **审计结果： 通过**

4.2.6 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**

4.2.7 整数溢出

- **风险描述**

整数溢出一般分为上溢和下溢，在智能合约中出现整数溢出的类型包括三种：乘法溢出、加法溢出、减法溢出。在 Solidity 语言中，变量支持的整数类型步长以 8 递增，支持从 uint8 到 uint256，以及 int8 到 int256，整数指定固定大小的数据类型，而且是无符号的，例如，一个 uint8 类型，只能存储在范围 0 到 2^8-1，也就是[0,255] 的数字，一个 uint256 类型，只能存储在范围 0 到 2^256-1 的数字。这意味着一个整型变量只能有一定范围的数字表示，不能超过这个制定的范围，超出变量类型所表达的数值范围将导致整数溢出漏洞。

- **审计结果： 通过**

4.2.8 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**

4.2.9 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**

4.2.10 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**

4.2.11 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**

4.2.12 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**

4.2.13 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**

4.2.14 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**

4.2.15 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**

4.2.16 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**

4.2.17 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**

4.2.18 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**

4.2.19 逻辑设计缺陷

- **漏洞描述**

在智能合约中，开发者为自己的合约设计的特殊功能意在稳固代币的市值或者项目的寿命，增加项目的亮点，然而越复杂的系统越容易有出错的可能，正是在这些逻辑和功能中，一个细微的失误就可能导致整个逻辑与预想出现严重的偏差，留下致命的隐患，比如逻辑判断错误，功能实现与设计不符等。

- **审计结果： 通过**

4.2.20 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**

4.2.21 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**

4.2.22 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**

4.2.23 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**

4.2.24 合约调用者未初始化

- **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**

4.2.25 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。

- **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