Cyberscope

Audit Report

Xocolatl HouseOfReserve

October 2022

Github	https://github.com/La-DAO/xocolatl-contracts
Commit	c367fec4a276bece4e580aca4a26e2147eb09643
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	HouseOfReserve
Compiler Version	v0.8.13+commit.abaa5c0e
Optimization	0 runs
Github	https://github.com/La-DAO/xocolatl-contracts/blob/main/contracts/HouseOfReserve.sol
Commit	c367fec4a276bece4e580aca4a26e2147eb09643
Testing Deploy	https://testnet.bscscan.com/token/0x09b99d80E2072712aEe8610F0d6CC12aDA4A03B7
Domain	https://xocolatl.club

Audit Updates

Initial Audit	21st October 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	5af1771388b4fe634e0a566716e32c6d00a5372875099127b274d4cf8a94e9d2
@openzeppelin/contracts/access/IAccessControl.sol	d03c1257f2094da6c86efa7aa09c1c07ebd33dd31046480c5097bc2542140e45
@openzeppelin/contracts/proxy/utils/Initializable.sol	36cf1b60e8da3e2bca15b187f775780310bb219c30dccd6258123c43fbf84ad8
@openzeppelin/contracts/token/ERC1155/IERC1155.sol	fd6a1801f1f2f8af0a3ece0b254da06ec24568aec02cfe94827061379aebc6f3
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/utils/Address.sol	1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/cryptography/ECDSA.sol	4e45d53327d561848fbcf381262ec5c0ac91b2f1f06432210bf76db55279d945
@openzeppelin/contracts/utils/introspection/ERC165.sol	8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5

@openzeppelin/contracts/utils/Strings.sol	34127ad0054df5963b0fd694c1b313d17e9114a2f426b85526d6d976210298ab
contracts/abstract/OracleHouse.sol	bda23986b2c82b00d3600c6b5ffaaccd2a46b8c0c5508fc97432fc5d9671341c
contracts/HouseOfReserve.sol	d235e2f37bb6a7494fd4ba323b8d74adc795d60eb58f81b28bbd434cfcc5dca3
contracts/interfaces/chainlink/IAggregatorV3.sol	299b7546616ad9fb756c778f0771f5d39aeca3f85fb2c4d794b19df0a8795bd3
contracts/interfaces/IAssetsAccountant.sol	9119e1160f73bf62a5ef77f66d6932615f52836ca70f66f3d5b82b59fe61b1e9
contracts/interfaces/IWETH.sol	aae423d3f0e5e6f0e62d62b6567ec2ec1a8965c70e2ffbd129f3d1e085ad941f
contracts/interfaces/uma/IAddressWhitelist.sol	46235463375dd715f5f30b2dd2bca0423e0994a311f84204ab39e82ef5d0e95b
contracts/interfaces/uma/IdentifierWhitelistInterface.sol	9495496b5ab855df3397193c9ba6a31eaf4ee050bce789bb2215619130723d3d
contracts/interfaces/uma/IOptimisticOracleV2.sol	11203bc5f10d2e4a60dcdb0f3728aae9f315bea16d5dbfa75fe6d5f0038f8aad
contracts/interfaces/uma/IUMAFinder.sol	94e604d5efcb6f22ea5f73d3c38c849775ae8225b9c736551db3d3cbaaa3bc93
contracts/utils/redstone/PriceAware.sol	0c7096448999fe38e17ca708ea0ad6dbb8878991413bfecfd09f4a1d7c7070b5
contracts/utils/uma/UMAOracleHelper.sol	d78c692b5c37e42e1d57ae6b8c6e08bda2a5db8e02d77ee46efecdb60ec422b1
contracts/utils/uma/UMAOracleInterfaces.sol	81eab927f79ea99651be5db8f7c3ae1fadaeed577a6b8ca53cc2c1cc77f3b55b

Introduction

The HouseOfReserve receives a reserved token in order to issue reserveTokenIds. The ratio between reserved and reserveTokenIds is 1-1. The funds are deposited to the HouseOfReserve contract. The mint is taking place on the AssetsAccountant contract.

The contract uses Oracles to receive off-chain data. Three oracles are configured Chainlink, Optimistic, and Redstone. The contract can use one Oracle at a time.

Roles

The admin role has the authority:

- To configure Oracles. The admin can activate, set tickers, set new oracle addresses, and authorize a new Signer to the Oracles. The owner is responsible for seting the proper tickers for the corresponding assets.
- To configure the deposit limit and the collateral ratio.
    - The collateral ratio is the ratio between the reserved and tha backed token.
    - The deposit limit controls the maximum total amount of reserve token that the contract accept.

Users can deposit and withdraw reserve tokens to the contract.

- Deposit, a user can deposit reserve tokens to the HouseOfReserve.
- Withdraw, a use can withdraw the reserved token. The withdraw amount depends on the backed tokens that have been issued.

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	STC	Succeeded Transfer Check	Unresolved
	MC	Missing Check	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L07	Missing Events Arithmetic	Unresolved
	L13	Divide before Multiply Operation	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L218,326
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IERC20(reserveAsset).transferFrom(msg.sender, address(this), amount);

IERC20(reserveAsset).transfer(msg.sender, amount);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L95
Status	Unresolved

Description

The contract is processing variables that have not been properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues. The contract doesn't sanitize the address on the initializer.

```
function initialize(
        address _reserveAsset,
        address _backedAsset,
        address _assetsAccountant,
        string memory tickerUsdFiat_,
        string memory tickerReserveAsset_,
        address _WETH
    ) public initializer {
        reserveAsset = _reserveAsset;
        backedAsset = _backedAsset;
        WETH = _WETH;
        reserveTokenID = uint256(
            keccak256(abi.encodePacked(reserveAsset, backedAsset, "collateral"))
        );
        backedTokenID = uint256(
            keccak256(
                abi.encodePacked(reserveAsset, backedAsset, "backedAsset")
            )
        );
        collateralRatio.numerator = 150;
        collateralRatio.denominator = 100;
        assetsAccountant = IAssetsAccountant(_assetsAccountant);
```

Recommendation

The contract should properly check the variables according to the required specifications. The addresses `_reserveAsset`, `_backedAsset`, `_assetsAccountant`, and `_WETH` should not be zero.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/HouseOfReserve.sol#L62,101,97,98,96
Status	Unresolved

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
WETH
_WETH
_backedAsset
_assetsAccountant
_reserveAsset
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contracts/HouseOfReserve.sol#L95
Status	Unresolved

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
reserveTokenID =
uint256(keccak256(bytes)(abi.encodePacked(reserveAsset,backedAsset,collateral)))
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contracts/HouseOfReserve.sol#L351
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
minReqReserveBal = (minReqReserveBal * collateralRatio.numerator) /
collateralRatio.denominator
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
Initializable	Implementation			
	_disableInitializers	Internal	✓	
IERC1155	Interface	IERC165		
	balanceOf	External		-

	balanceOfBatch	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
	safeBatchTransferFrom	External	✓	-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ECDSA	Library			
	_throwError	Private		
	tryRecover	Internal		
	recover	Internal		

	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	toEthSignedMessageHash	Internal		
	toEthSignedMessageHash	Internal		
	toTypedDataHash	Internal		
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC165	Interface			
	supportsInterface	External		-
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
OracleHouse	Implementation	PriceAware		
	_oracleHouse_init	Internal	✓	
	activeOracle	External		-
	_getLatestPrice	Internal		
	setActiveOracle	External	✓	-
	_setActiveOracle	Internal	✓	
	_oracle_redstone_init	Private	✓	
	_getLatestPriceRedstone	Internal		
	getRedstoneData	External		-
	isSignerAuthorized	Public		-
	setTickers	External	✓	-
	_setTickers	Internal	✓	
	authorizeSigner	External	✓	-
	_authorizeSigner	Internal	✓	
	_getLatestPriceUMA	Internal		

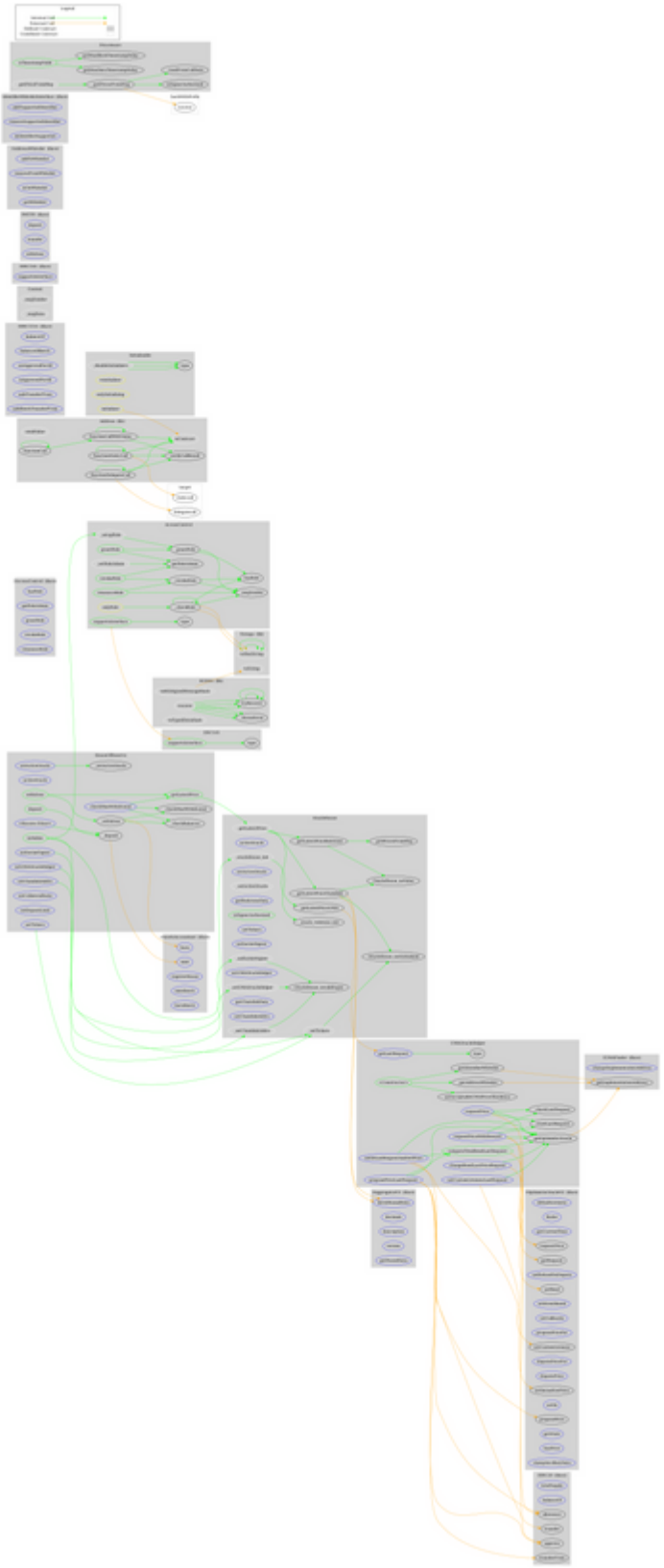
	setUMAOracleHelper	External	✓	-
	_setUMAOracleHelper	Internal	✓	
	_getLatestPriceChainlink	Internal		
	getChainlinkData	External		-
	setChainlinkAddrs	External	✓	-
	_setChainlinkAddrs	Internal	✓	
HouseOfReserveState	Implementation			
HouseOfReserve	Implementation	Initializable, AccessControl, OracleHouse, HouseOfReserveState		
	initialize	Public	✓	initializer
	activeOracle	External		-
	setActiveOracle	External	✓	onlyRole
	setTickers	External	✓	onlyRole
	authorizeSigner	External	✓	onlyRole
	setUMAOracleHelper	External	✓	onlyRole
	setChainlinkAddrs	External	✓	onlyRole
	getLatestPrice	Public		-
	deposit	Public	✓	-
	withdraw	Public	✓	-
	setCollateralRatio	External	✓	onlyRole
	setDepositLimit	External	✓	onlyRole
	checkMaxWithdrawal	External		-
	_withdraw	Internal	✓	
	_deposit	Internal	✓	
	_checkMaxWithdrawal	Internal		
	_checkBalances	Internal		
	<Receive Ether>	External	Payable	-
IAggregatorV3	Interface			

	decimals	External		-
	description	External		-
	version	External		-
	getRoundData	External		-
	latestRoundData	External		-
IAssetsAccountant	Interface	IERC1155		
	registerHouse	External	✓	-
	mint	External	✓	-
	mintBatch	External	✓	-
	burn	External	✓	-
	burnBatch	External	✓	-
IWETH	Interface			
	deposit	External	Payable	-
	transfer	External	✓	-
	withdraw	External	✓	-
IAddressWhitelist	Interface			
	addToWhitelist	External	✓	-
	removeFromWhitelist	External	✓	-
	isOnWhitelist	External		-
	getWhitelist	External		-
IdentifierWhitelistInterface	Interface			
	addSupportedIdentifier	External	✓	-
	removeSupportedIdentifier	External	✓	-
	isIdentifierSupported	External		-
IOptimisticOracleV2	Interface			
	defaultLiveness	External		-
	finder	External		-

	getCurrentTime	External		-
	requestPrice	External	✓	-
	setBond	External	✓	-
	setRefundOnDispute	External	✓	-
	setCustomLiveness	External	✓	-
	setEventBased	External	✓	-
	setCallbacks	External	✓	-
	proposePriceFor	External	✓	-
	proposePrice	External	✓	-
	disputePriceFor	External	✓	-
	disputePrice	External	✓	-
	settleAndGetPrice	External	✓	-
	settle	External	✓	-
	getRequest	External		-
	getState	External		-
	hasPrice	External		-
	stampAncillaryData	External		-
IUMAFinder	Interface			
	changeImplementationAddress	External	✓	-
	getImplementationAddress	External		-
PriceAware	Implementation			
	getMaxDataTimestampDelay	Public		-
	getMaxBlockTimestampDelay	Public		-
	isSignerAuthorized	Public		-
	isTimestampValid	Public		-
	_getPriceFromMsg	Internal		
	_getPricesFromMsg	Internal		
	_readFromCallData	Private		
UMAOracleHelper	Implementation			
	<Constructor>	Public	✓	-
	getLastRequest	External		-

	requestPrice	External	✓	-
	requestPriceWithReward	External	✓	-
	setCustomLivenessLastRequest	External	✓	-
	changeBondLastPriceRequest	External	✓	-
	computeTotalBondLastRequest	Public		-
	proposePriceLastRequest	External	✓	-
	settleLastRequestAndGetPrice	External	✓	-
	setAcceptableUMAPriceObsolence	Public	✓	-
	_checkLastRequest	Internal		
	_resetLastRequest	Internal	✓	
	_getIdentifierWhitelist	Internal		
	_getAddressWhitelist	Internal		
	_getOptimisticOracle	Internal		
UMAOracleInterfaces	Library			

Contract Flow



Domain Info

Domain Name	xocolatl.club
Registry Domain ID	D017C2E7D305043B48BB9BAC3CE267A07-GDREG
Creation Date	2022-09-09T07:58:44Z
Updated Date	2022-09-14T07:58:44Z
Registry Expiry Date	2023-09-09T07:58:44Z
Registrar WHOIS Server	whois.opensrs.net
Registrar URL	www.opensrs.com
Registrar	Tucows Domains Inc.
Registrar IANA ID	69

The domain was created about 1 month before the creation of the audit. It will expire in 11 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

The HouseOfReserve contract implements a collateral issuing mechanism. This audit investigates security issues and mentions business logic concerns and potential improvements.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io