Cyberscope

Audit Report

Xocolatl HouseOfCoin

October 2022

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

Table of Contents

Contract Review

Contract Name	HouseOfCoin
Compiler Version	v0.8.13+commit.abaa5c0e
Optimization	0 runs
Github	https://github.com/La-DAO/xocolatl-contracts/blob/main/contracts/HouseOfCoin.sol
Commit	c367fec4a276bece4e580aca4a26e2147eb09643
Testing Deploy	https://testnet.bscscan.com/token/0xCb31c74596647c714cC7D46166017Cac206fdec0
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/HouseOfCoin.sol	69d62809a6ba6973bc05a8a3f7e7400362ff381ce782642f7e1d0829910125ae
contracts/interfaces/chainlink/IAggregatorV3.sol	299b7546616ad9fb756c778f0771f5d39aeca3f85fb2c4d794b19df0a8795bd3
contracts/interfaces/IAssetsAccountant.sol	9119e1160f73bf62a5ef77f66d6932615f52836ca70f66f3d5b82b59fe61b1e9
contracts/interfaces/IERC20Extension.sol	341c5d7640bd0c44aa86ec924574727c53604487e57352158fb9a11e3b671f8d
contracts/interfaces/IHouseOfReserve.sol	2cf3c1454c96809fe84a571802268e15539652ab80328dbc7cd99b1db5f7997e
contracts/interfaces/IOracle.sol	1f13347804c9d374a356eb2c5100a4f983c3873c164e5bd1d3890d79bc3786a4
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 HouseOfCoin is a collateral issuing mechanim. The contract is responsible for minting backed tokens is relation to the reserved tokens. The ratio between the backed and the reserved tokens is defined by two factors, the price and the collarationRatio. The price is defined by an oracle and the collarationRatio by the admin role.

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 authority:

- Configure the Oracles.
- Configure the liquidation parameters.

Users has the ability to:

- Mint coins. Issue backed tokens proportionally to their reserved tokens.
- Payback coins. Burn backed their tokens.
- Liquidate. Get the reserves of a user by burning the caller’s backed tokens.
- Chck the user health ratio.
- Check the cost of liquidation.
- Check the remaining minting power.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	MCM	Mint Coin Mocking	Unresolved
●	LP	Liquidate Permissions	Unresolved
●	MC	Missing Check	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

MC - Mint Coin Mocking

Criticality	critical
Location	contract.sol#L260
Status	Unresolved

Description

The caller has the ability to mock the mintCoin parameters in order to mint unlimited amount of backed tokens. The caller can create a custom contract that behave similarly to a “houseOfReserve” contract. Steps to reproduce:

1. The user issues reserved tokens from the HouseOfReserve. For instance: `deposit(1)`.
2. The user creates a contract that implements the same API with the HouseOfReserve.
    - 2.1. `reserveTokenID()` will return the actual reserveTokenID
    - 2.2. `reserveAsset` will return the actual reserveAsset address
    - 2.3. `collateralRatio()` will return a high value, for instance 100^18.
    - 2.4. The `activeOracle` could return 1 and `getLatestPrice()` an normal price.
3. The user calls the mintCoin method providing the actual reserveAsset, the mocked HouseOfReserve and a huge amount. For instance: `mintCoin(reserveAsset, mockedHouseOfReserve, 100^12)`

```
function mintCoin(
    address reserveAsset,
    address houseOfReserve,
```

```
    uint256 amount
) public returns (bool success) {
...
}
```

Recommendation

The contract should accept parameters only from approved addresses that cannot be manipulated.

LP - Liquidate Permissions

Criticality	minor / informative
Location	contract.sol#L361
Status	Unresolved

Description

Any user has the authority to call the ‘liquidateUser()’ method. There are two cases:

- If the targeted user health ratio is between ‘marginCallThreshold’ and ‘liquidationThreshold’, then it will emit a ‘MarginCall’ event.
- If the health ratio is less than ‘liquidationThreshold’ and the callers role is not LIQUIDATOR, then the contract will abort the transaction, otherwise it will proceed.

This diversion may produce some misconserns since any user can emit the event but specific users can proceed with the liquidation.

```
function liquidateUser(address userToLiquidate, address reserveAsset)
    external
{
...
}
```

Recommendation

The contract could allow only the LIQUIDATOR role to access the ‘liquidateUser()’ method.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol#L132
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 _backedAsset, address _assetsAccountant)
        public
        initializer
    {
        backedAsset = _backedAsset;
        backedAssetDecimals = IERC20Extension(backedAsset).decimals();
        assetsAccountant = _assetsAccountant;
```

Recommendation

The contract should properly check the variables according to the required specifications. The address `_assetsAccountant` should not be set to zero address.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/HouseOfCoin.sol#L559,561,563,522,562,327,132,560,118
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.

```
_globalBase
_liquidationThreshold
_collateralPenalty
_reserveAsset
_liquidationPricePenaltyDiscount
_backedTokenID
_assetsAccountant
_backedAsset
_marginCallThreshold
...
```

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/HouseOfCoin.sol#L132
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.

```
backedAssetDecimals = IERC20Extension(backedAsset).decimals()
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contracts/HouseOfCoin.sol#L741,700,762
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
reserveBalreducedByFactor = (reserveBal * collatRatio.denominator) / collatRatio.numerator
liqDiscountedPrice = (price * discount) / _liqParam.globalBase
amountTemp = amountTemp / 10 ** decimalDiff
maxMintableAmount = (reserveBalreducedByFactor * price) / 1e8
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contracts/HouseOfCoin.sol#L722
Status	Unresolved

Description

The are variables that are defined in the local scope and are not initialized.

```
Itemp
```

Recommendation

All the local scoped variables should be initialized.

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	✓	
HouseOfCoinState	Implementation			
HouseOfCoin	Implementation	Initializable, AccessControl, OracleHouse, HouseOfCoinState		
	initialize	Public	✓	initializer
	activeOracle	External		-
	setActiveOracle	External		-
	setTickers	External		-
	getRedstoneData	External		-
	authorizeSigner	External	✓	onlyRole
	setUMAOracleHelper	External		-
	getChainlinkData	External		-
	setChainlinkAddrs	External		-
	getLatestPrice	Public		-
	_getLatestPrice	Internal		
	mintCoin	Public	✓	-
	paybackCoin	Public	✓	-
	liquidateUser	External	✓	-
	computeUserHealthRatio	Public		-
	computeCostOfLiquidation	Public		-
	getBackedTokenID	Public		-
	getLiqParams	Public		-
	setLiqParams	Public	✓	onlyRole
	checkRemainingMintingPower	Public		-

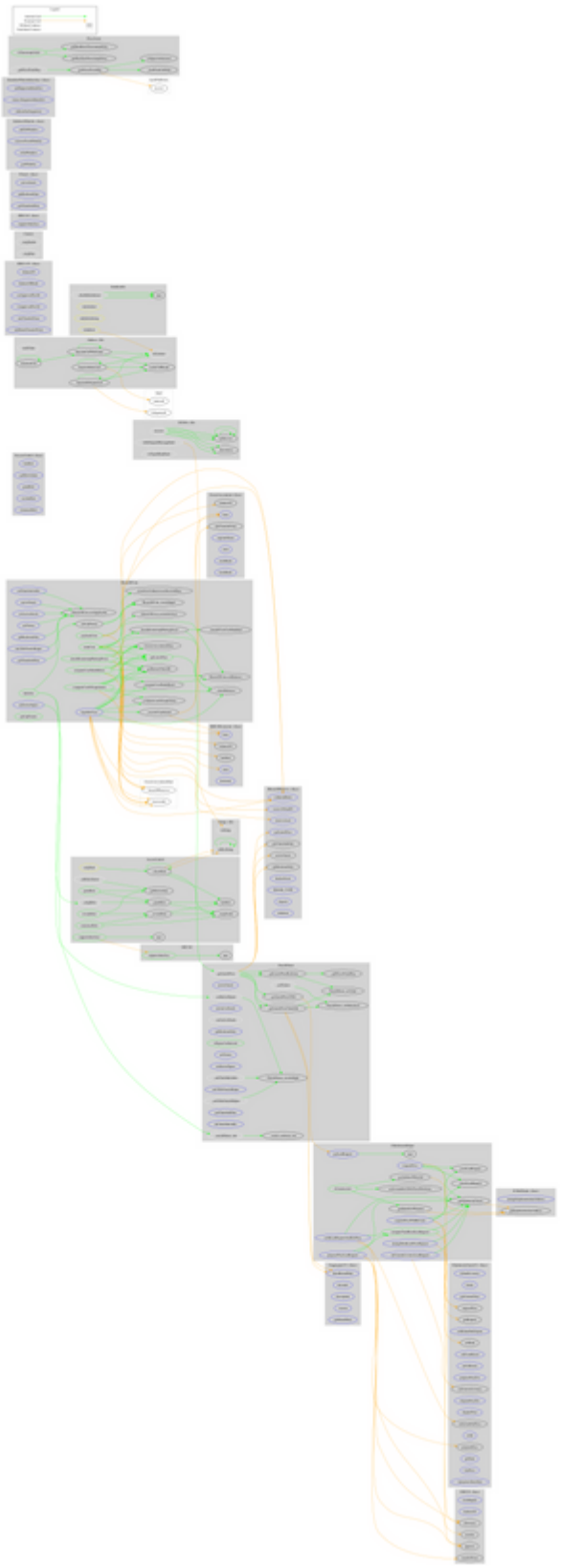
	_checkBalances	Internal		
	_checkRemainingMintingPower	Internal		
	_checkIfUserCanMintMore	Internal		
	_transformToBackAssetDecimalBase	Internal	✓	
	_computeUserHealthRatio	Internal		
	_computeCostOfLiquidation	Internal		
	_executeLiquidation	Internal	✓	
IAggregatorV3	Interface			
	decimals	External		-
	description	External		-
	version	External		-
	getRoundData	External		-
	latestRoundData	External		-
IAssetsAccountant	Interface	IERC1155		
	registerHouse	External	✓	-
	mint	External	✓	-
	mintBatch	External	✓	-
	burn	External	✓	-
	burnBatch	External	✓	-
IERC20Extension	Interface	IERC20, IAccessControl		
	decimals	External		-
	mint	External	✓	-
	burn	External	✓	-
IHouseOfReserve	Interface	IOracle		
	reserveAsset	External		-
	backedAsset	External		-
	reserveTokenID	External		-
	HOUSE_TYPE	External	✓	-

	collateralRatio	External		-
	getLatestPrice	External		-
	deposit	External	✓	-
	withdraw	External	✓	-
IOracle	Interface			
	activeOracle	External		-
	getRedstoneData	External		-
	getChainlinkData	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 HouseOfCoin is a collateral issuing mechanim. 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