Cyberscope

Audit Report

Xocolatl HouseOfReserve

December 2022

Github	https://github.com/La-DAO/xocolatl-contracts
Commit	7d780e9a7573b88f042f8f45096a201442ea782e
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	HouseOfReserve
Testing Deploy	https://testnet.bscscan.com/address/0xfe5d38fca4560209d99a67fe622b02c6cb086793

Audit Updates

Initial Audit	24 Oct 2022 https://github.com/cyberscope-io/audits/blob/main/xocolatl/v1/houseOfReserve.pdf
Corrected Phase 2	19 Dec 2022

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	525910346dd07b0a3e1950dc676938240bf4b9c48b1d3f94f9f6b361ea197a2b
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 setting 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 the backed token.
    - The deposit limit controls the maximum total amount of reserve token that the contract accepts.

Users can deposit and withdraw reserve tokens to the contract.

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

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L16	Validate Variable Setters	Unresolved
●	L18	Multiple Pragma Directives	Unresolved
●	L20	Succeeded Transfer Check	Unresolved

MC - Missing Check

Criticality	Minor / Informative
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.

```
function initialize(
    address _reserveAsset,
    address _backedAsset,
    address _assetsAccountant,
    string memory tickerUsdFiat_,
    string memory tickerReserveAsset_,
    address _WETH
) public initializer {
    reserveAsset = _reserveAsset;
    backedAsset = _backedAsset;
    WETH = _WETH;
    ...
```

Recommendation

The team is advised to 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#L71,105,106,107,110
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of your Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of your code.

7. Keep lines short (around 120 characters) to improve readability.

```
address public WETH
address _reserveAsset
address _backedAsset
address _assetsAccountant

address _WETH
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

You can find more information on the Solidity documentation https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

L07 - Missing Events Arithmetic

Criticality	Minor / Informative
Location	contracts/HouseOfReserve.sol#L115
Status	Unresolved

Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task.

It's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

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

Recommendation

By including all required events in the contract and thoroughly testing the contract's functionality, you can help to ensure that the contract performs as intended and does not have any missing events that could cause issues with its arithmetic.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	contracts/HouseOfReserve.sol#L112,113,114
Status	Unresolved

Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
reserveAsset = _reserveAsset
backedAsset = _backedAsset

WETH = _WETH
```

Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

L18 - Multiple Pragma Directives

Criticality	Minor / Informative
Location	contracts/HouseOfReserve.sol#L2
Status	Unresolved

Description

If the contract includes multiple conflicting pragma directives, it may produce unexpected errors. To avoid this, it's important to include the correct pragma directive at the top of the contract and to ensure that it is the only pragma directive included in the contract.

```
pragma solidity 0.8.13;
```

Recommendation

It is important to include only one pragma directive at the top of the contract and to ensure that it accurately reflects the version of Solidity that the contract is written in. By including all required compiler options and flags in a single pragma directive, you can avoid conflicts and ensure that the contract can be compiled correctly.

L20 - Succeeded Transfer Check

Criticality	Minor / Informative
Location	contracts/HouseOfReserve.sol#L243,351
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. The team is advised to check the SafeERC20 library from the Openzeppelin library.

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		

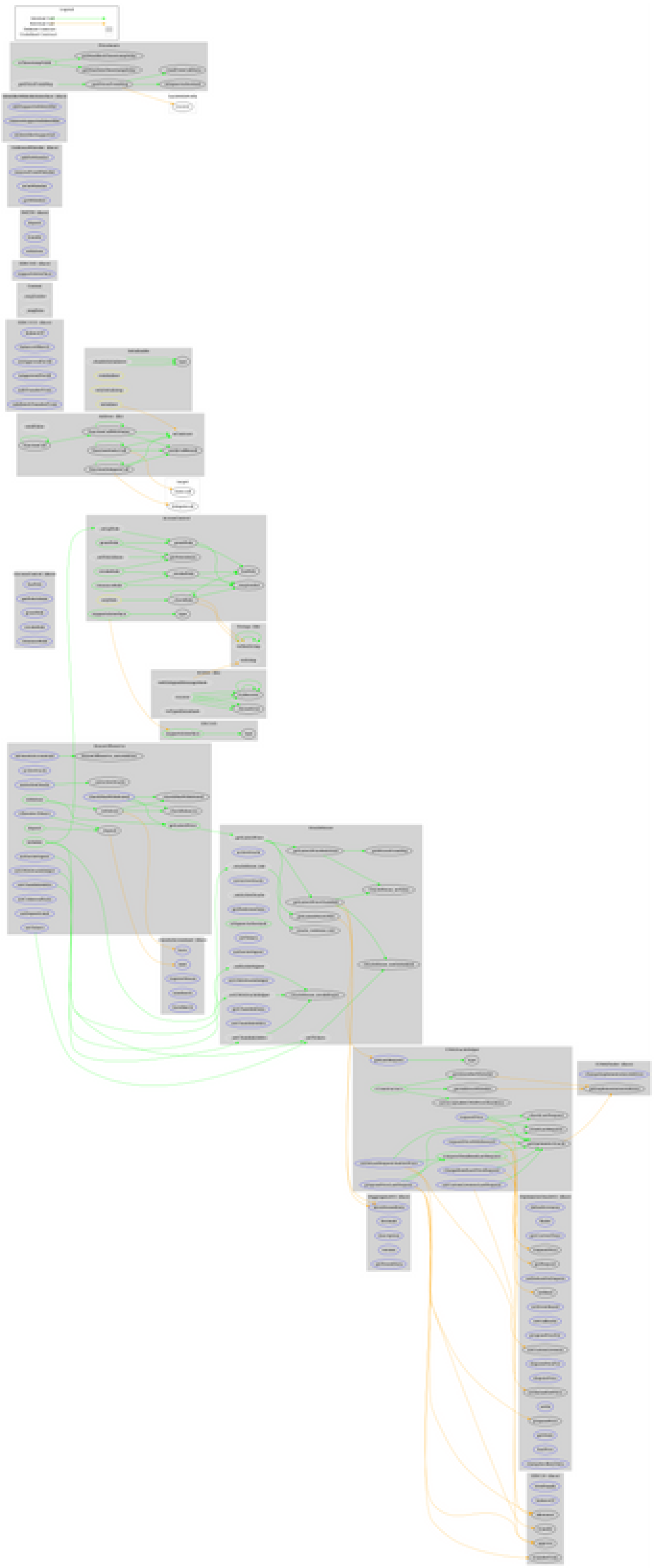
	initialize	Public	✓	initializer
	activeOracle	External		-
	setAssetsAccountant	External	✓	onlyRole
	setActiveOracle	External	✓	onlyRole
	setTickers	External	✓	onlyRole
	authorizeSigner	External	✓	onlyRole
	setUMAOracleHelper	External	✓	onlyRole
	setChainlinkAddrs	External	✓	onlyRole
	getLatestPrice	Public		-
	deposit	Public	✓	-
	withdraw	Public	✓	-
	setCollateralRatio	External	✓	onlyRole
	setDepositLimit	External	✓	onlyRole
	checkMaxWithdrawal	External		-
	_withdraw	Internal	✓	
	_deposit	Internal	✓	
	_checkMaxWithdrawal	Internal		
	_checkBalances	Internal		
		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	✓	-

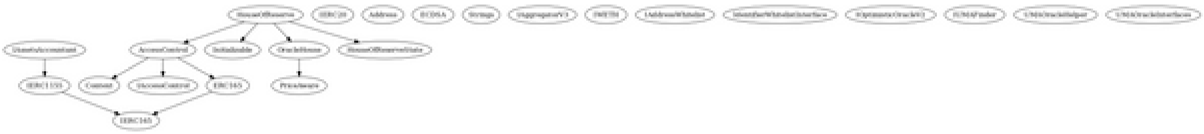
	setCustomLiveness	External	✓	-
	setEventBased	External	✓	-
	setCallbacks	External	✓	-
	proposePriceFor	External	✓	-
	proposePrice	External	✓	-
	disputePriceFor	External	✓	-
	disputePrice	External	✓	-
	settleAndGetPrice	External	✓	-
	settle	External	✓	-
	getRequest	External		-
	getState	External		-
	hasPrice	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			
		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



Inheritance Graph



Summary

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

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io