ExOracle

合約審計報告

VER 1.0

2022年5月24日

No. 2022052411230

項目總結

1. 項目介紹

ExOracle 是一個OKC的官方鏈上預言機。為OKC鏈上的Dapp項目提供各類鏈下數據，包括但不限於BTC-USD等各種數字資產的價格。鏈上其他項目可以與ExOracle交互，獲取最新的相關價格數據。價格提交者還可以獲得一些獎勵。

每個Dapp合約與EXO集成時，都需要指定一些特定的地址源地址（source）來獲取價格，以防數據從未知地址發布。EXOracle在各類文檔中引導用戶盡量使用EXOracle官方source，以獲得及時準確的價格服務。

ExOracle還借助鏈上Dex來實現價格容錯機製，旨在對後端提供的報價進行一個兜底檢驗：合約會對後端傳入的價格數據做一個價格偏移判斷。

具體做法：判斷合約新價格與合約上一次存儲的價格的偏差是否處於規定的範圍內。

2. 審計詳情

項目名稱	ExOracle	平台	N/A
通證名稱	N/A	通證代號	N/A
開始時間	2022年5月30日	語言	Solidity
結束時間	2022年6月2日	官網	N/A
Github	https://github.com/okex/OKCOracle/tree/0b0a8061e96256fa0773cf5ec05424417c251897	白皮書	N/A

2022年6月6日

3. 審計範圍

ID	文件	SHA-256 checksum
contracts	DEXOracleFactory.sol	8d6bf86356327036f75b102bf698550a0b4ce747dba890600a951cd30b280b5b
contracts	DEXOracleSimple.sol	c422215ed4ce0f70f547f45024ad25ee48e2a96e88767e098a8d4f95a82bf0c2
contracts	ExOracle.sol	e67f06fe7217717b1c10c2f3f2443a7063279c471ab28d0770208a191fcad0bd
contracts	OwnedUpgradeabilityProxy.sol	95f72bd8101f6510e5e4bbf22afa3eece678a9f98f42cfb4fce42c56a23f5cd7
contracts	PriceCumulativeData.sol	6846ce6fdada73f15154afa8ca7266086d4d737c01010f157b4e77240955347b
contracts	RequestViewList.sol	ec96bdd4390e20be4d403b7568c3c2961b2be621820fc023c24fb3020859e18f
contracts/libraries	Address.sol	daec46987225cfcac1f7aa7fa8728f4b306ae84e9702f809aa0982dca0bd1523
contracts/libraries	EnumerableSet.sol	2820521c4e2adcccbb2d12c1e82ff3946b474be1ad0dc2d2e514b8da2d0ac617
contracts/libraries	Math.sol	534e4353a4e96ae4097f81e1a620faa5464222db5bb9d9b567a92f58d31813fe
contracts/libraries	Median.sol	a48fdf7b6200dee11a9256e893c68a76f4fda5b5b0be44a36689c0e1933d0577
contracts/libraries	SafeERC20.sol	8b2244ca44e2e051bea1cedfede5efa9ae1ddcee91320bcffdb78268d7fa43dd
contracts/libraries	SafeMath.sol	5f5fe35a58b19d336ea08413d6ecec0a11d1fee7674c666528107e80e71135a8
contracts/libraries	SignedSafeMath.sol	5f305a9e2a0fef313ee6d1fbd10af284e30632f78402e059800e76480c4a01db
contracts/libraries	UniswapV2OracleLibrary.sol	3e62dd20c143244030b0c5845d7d696b74c64519c4b41b28e68a9504f957e8db
contracts/utils	AutoExtendArrayAddress.sol	a332d27f524825a0263a6eb663fe98ab3a117e6e8320558139167b040434d944
contracts/utils	AutoExtendArrayInt.sol	37afed952cfa633346b03b5bffd5c33bc4f7a955084586100d3387b026ebc44d
contracts/utils	Context.sol	6a25312554a817075fbe85e3c57f5e0ecad5b0bac4303bd3967ce9680e71a2af
contracts/utils	ERC165.sol	87ca2102c785b8d030492d1d290ccf52f7352bb6090dcf87d39ade34651a7bbf
contracts/utils	ERC20.sol	537854f9f79166df0accdb75c8a7a3309354dfe35e8f838fcd8d9c223c880b99
contracts/utils	ERC20Burnable.sol	d3ea637d1b59d04ad5af29d2a6de57d7bd24cd3142c3d6e95b2446bd60e2787f
contracts/utils	Owned.sol	8d75e85114f3b4da66cb7db2cceaf95ee70cf2209a8d5daadef9f2a6996f6374

2022年6月6日

4. 代碼結構

```
├── DEXOracleFactory.sol #DEXOracle工厂，可创建DEXOracleSimple合约，并管理所创建合约的数据
├── DEXOracleSimple.sol #获取DEX交易对价格
├── ExOracle.sol #喂价和获取价格
├── OwnedUpgradeabilityProxy.sol #可升级proxy合约
├── PriceCumulativeData.sol #累积价格计算
├── RequestViewList.sol #用户request view列表
├── interfaces #接口
│   ├── IAccessControllerInterface.sol
│   ├── IDexOracleFactory.sol
│   ├── IDexOracleSimple.sol
│   ├── IERC165.sol
│   ├── IERC20.sol
│   ├── IERC20Metadata.sol
│   ├── IExOraclePriceData.sol
│   ├── IPriceCumulative.sol
│   └── IRequesterView.sol
├── libraries
│   ├── Address.sol #Address库
│   ├── EnumerableSet.sol #可枚举set库
│   ├── Math.sol #计算库
│   ├── Median.sol #求平均
│   ├── SafeERC20.sol #安全ERC20库
│   ├── SafeMath.sol #安全计算库
│   ├── SignedSafeMath.sol #有符号安全计算库
│   └── UniswapV2OracleLibrary.sol #uni v2 TWAP库
└── utils
    ├── AutoExtendArrayAddress.sol #自增地址数组
    ├── AutoExtendArrayInt.sol #自增int数组
    ├── Context.sol #msg上下文处理
    ├── ERC165.sol #ERC165标准
    ├── ERC20.sol #ERC20标准
    ├── ERC20Burnable.sol #可燃烧ERC20
    └── Owned.sol #owner控制
```

2022年6月6日

審計報告匯總

1. 審計方式

通過清晰地理解該項目的設計目的、運行原理和實現管道，稽核團隊對合約程式碼進行了深入的研究和分析。 在分清各個合約及其函數的調用關係的基礎上，對合約可能存在的漏洞進行了定位及分析。 最終產生問題描述和給出相應的修改意見。

審計方法	Static analysis, Manual Review

2. 審計流程

步驟	操作	詳細內容
1	背景研究	閱讀項目介紹、白皮書、合約源碼等項目方團隊提供的相關信息，確保正確理解項目功能
2	自動化檢測	主要用自動化工具掃描源碼，找到常見的潛在漏洞
3	人工審閱	工程師逐行閱讀代碼，找到潛在漏洞
4	邏輯校對	工程師將對代碼的理解和項目方提供的信息比較，檢查代碼實現是否符合項目白皮書信息
5	測試用例檢測	包括測試用例設計，測試範圍分析、符號執行等
6	優化審查	根據應用場景、調用方式及最新的研究成果從可維護性、安全性及可操作性等方面審查項目

2022年6月6日

3. 風險分級

風險級別	風險描述
致命	存在致命風險及隱患，需要立即解決
高危	存在高危風險及隱患，將引發相同問題，必須解決
中度	存在中度風險及隱患，可能導致潛在風險，最終仍然需要解決
低危	存在低風險及隱患，指各類處理不當或會引發警告信息的細節，這類問題可暫時擱置
建議	存在可優化的部分，這類問題可以擱置，但建議最終解決

4. 審計結果

編號	審計項目	風險級別	狀態
1	重入	無	
2	注入	無	
3	權限繞過	無	
4	Mempool搶跑	無	
5	回滾	無	
6	條件競爭	無	
7	循環耗盡gas	中	已告知
8	閃電貸高影響	無	
9	經濟模型不合理	無	
10	可預見的隨機數	無	
11	投票權管理混亂	無	

2022年6月6日

編號	審計項目	風險級別	狀態
12	數據隱私洩露	無	
13	鏈上時間使用不當	無	
14	Fallback函數編碼不當	無	
15	鑒權不當	無	
16	Opcode使用不當	無	
17	內聯匯編使用不當	無	
18	構造函數不規範	無	
19	返回值不規範	無	
20	Event不規範	無	
21	關鍵字使用不規範	無	
22	未遵循ERC標準	無	
23	條件判斷不規範	無	
24	流動性枯竭風險	無	
25	中心化風險	高	已告知
26	邏輯變更風險	無	
27	整數溢出	無	
28	函數可見性不當	無	
29	變量初始化不當	無	
30	合約間調用不當	無	
31	變量不規範	無	
32	重放	無	
33	隨機存儲位置寫入	無	
34	蜜罐邏輯	無	
35	哈希碰撞	無	
36	使用不推薦的方法	無	
37	未遵循基本編碼原則	無	
38	第三方依賴風險	無	

上述表格中，狀態欄內容若為「已告知」，則表示審計團隊已告知項目方項目存在的漏洞，但項目方未對漏洞進行修改，或未告知審計團隊漏洞的修改進度。若狀態欄中填寫「已修改」則表示項目方已進行對漏洞的修改，並通過審計團隊確認。

2022年6月6日

5. 風險項與修改方案

以下部分為審計後得知的風險項相關詳細信息，其中內容包括風險類型、風險級別、問題位置、問題描述、修改建議及項目方反饋。

1. 循环耗尽gas

位置	文件	風險状态	風險級別
Line 615、651	ExOracle.sol	⚠️ 已告知	中风险

① 风险描述

以太坊網絡限定了每個區塊的最大 gas 總量值，區塊中所有交易的 gas 總和不能超過此區塊最大 gas 總量值。一旦合約中的某個操作將大量消耗 gas 以至於消耗的 gas 值達到了區塊最大 gas 總量值，此操作將不會被成功執行，所有依賴此操作的驗證步驟也將失效，合約會因此無法正常完成其余功能，從而造成一種拒絕服務狀態。
通常當一個合約開發者未考慮到區塊 gasLimit 而在合約中引入了修改隨時間增加大小會改變的數組等動態數據結構變量操作時，會發生此種拒絕服務攻擊。
攻擊者可以在一個區塊被開采出來後，馬上以較高的 gas 價格發出多個交易，然後利用合約上述操作消耗整個區塊 gas 限額，使該區塊在特定的時間之前不包含其他任何交易，以阻止其他用戶正常使用合約的功能，從而達到阻止正常提交交易的目標。
具體到該項目，可能會導致source地址正常更新價格受阻。

② 修改建议

循環未限製次數，且循環內存在外部調用，雖然該函數主要由OKC官方調用，但仍然存在被其他地址調用的可能，**建議限定循環次數**。

2022年6月6日

③ 关联代码 1

```
 /**
     * @notice function is supposed to be called by the price feeders when
they are feeding data to the oracle.
     *         This put function accepts multiple priceTypes at the same time.
     */
    function put(bytes[] calldata messages, bytes[] calldata signatures)
external  {
        //@Description: 循环未限制次数，且循环内存在外部调用，虽然该函数主要由OKC官方调
用，但仍然存在被其他地址调用的可能
        //@Solution: 限定循环次数
        for (uint256 i; i < signatures.length; i++)
        {
            (address source, uint64 value, string memory priceType, uint64
timestamp) = decodeMessage(messages[i], signatures[i]);
            lastResponseTime[source] = block.timestamp;
            source = checkFailOver(source);
            bool result = putInternal(priceType, source, value, timestamp);
            if(result){
                if(recipients[msg.sender] != address(0)){
                    credits[recipients[msg.sender]] =
credits[recipients[msg.sender]].add(creditIncrementNumber);
                }else{
                    credits[msg.sender] =
credits[msg.sender].add(creditIncrementNumber);
                }
            }
        }
    }
```

2022年6月6日

③ 关联代码 2

```
    /**
     * @notice postMining for the same requester with multiple different
tasks.
     */
    function postMiningMessages(address requester, bytes[] calldata messages,
bytes[] calldata signatures) external
    {
        require(messages.length == signatures.length, "length of meesages
should be the same as the length of signatures");
        //@Description: 循环未限制次数，且循环内存在外部调用
        //@Solution: 限定循环次数
        for (uint256 i; i < messages.length; i++)
        {
            (address source, uint64 timestamp, string memory priceType,
uint64 value) = decodeMessage(messages[i], signatures[i]);
            lastResponseTime[source] = block.timestamp;
            source = checkFailOver(source);
            postMiningInternal(requester, source, timestamp, priceType,
value);
        }
    }
```

2022年6月6日

2. 循环耗尽gas

位置	文件	風險状态	風險級别
Line 470	ExOracle.sol	⚠️ 已告知	高风险

① 风险描述

中心化風險是一類由於權限控製或關鍵合約操作由單一私鑰控製的風險。由於使用單一私鑰，在極端情況下，有可能出現私鑰丟失或私鑰泄露的風險。一旦私鑰丟失，則合約的關鍵管理功能將無法使用，而私鑰泄露則會造成更嚴重的資金風險。

一般建議權限控製和關鍵的合約管理功能使用多簽或timelock進行操作，而對於有中心化服務器進行自動化操作的功能，需要項目方建立好故障報警和修復機製，以及時處理不可預期的網絡情況。

該項目的中心化風險除了在owner權限的單一私鑰問題外，還在於餵價機製中的dex兜底機製：由於餵價最後要和dex價格做比較，相差超過一定閾值後不予更新，會導致在dex更新價格的時間內（默認4小時），極端行情下，餵價機製可能失敗。需要OKC官方及時發現故障，並調整dex更新時間或閾值。

② 修改建议

與dex價格進行比較，dex價格更新時間和比較閾值需要在極端行情下手動更新，**推荐建立好故障報警和修復機製。**

2022年6月6日

③ 关联代码 1

```
 /**
     * @dev putInternal handles the logic when a price is requested to be
update. Only update when the submitted value is at a newer timestamp
     */
    function putInternal(string memory priceType, address source,  uint64
timestamp, uint64 value) internal checkAllowedList(priceType, source) returns
(bool) {
        // Only update if newer than stored, according to source
        Datum storage prior = data[priceType][source];
        //add deadline for value
        if (timestamp > prior.timestamp && source != address(0) ) {
            require(checkTimestamp(timestamp),"timestamp is invalid");
            Datum memory newPrice = Datum(timestamp, value);
            uint256 priceDeviation = priceDeviationThreshold[priceType];
            bool isCheck = isCheckDexPrice[priceType];  //default is false
            //default priceDeviationThreshold is 50%
            if(priceDeviation == 0){
                priceDeviationThreshold[priceType] =
DEFAULT_PRICEDEVIATION_THRESHOLD;
                priceDeviation = priceDeviationThreshold[priceType];
            }
            if(priceChangeAboveMax(newPrice,prior,priceDeviation)){
                emit PriceChangeAboveMax(newPrice,prior);
                emit NotWritten(source, priceType, data[priceType]
[source].timestamp, timestamp, block.timestamp);
                return false;
            }
```

2022年6月6日

③ 关联代码 1

```
            if(isCheck){
            //@Description: 与dex价格进行比较，dex价格更新时间和比较阈值需要在极端行情下手
动更新
            //@Solution: 建立好故障报警和修复机制
                if(!dexOracleFactory.isPriceSimilar(priceType,
uint256(value))){
                    emit DexPriceMisMatch(priceType,value);
                    emit NotWritten(source, priceType, data[priceType]
[source].timestamp, timestamp, block.timestamp);
                    return false;
                }
            }
            data[priceType][source] = newPrice;
            priceCumulative.priceCumulative(priceType,source,value);
            emit Write(source, priceType, timestamp, value);
            return true;
        } else {
            emit NotWritten(source, priceType, data[priceType]
[source].timestamp, timestamp, block.timestamp);
            return false;
        }
    }
```

2022年6月6日

免責聲明

i. 本審計報告僅針對最終出具報告中載明的審計類型進行審計，其他未知安全漏洞不在本次審計責任範圍之內，我方無需為此承擔責任。

ii. 我方僅應根據審計報告發布之前存在或發生的攻擊或漏洞發布審計報告。對於將來存在或發生的新攻擊或漏洞，我方無法確定對其項目安全狀態的可能影響，對此概不負責。

iii. 我方發布的審計報告中的安全審計分析及其他內容應僅基於項目方在發布審計報告之前向我方提供的文件和材料（包括但不限於合約代碼），並且上述文件和資料不應該存在缺乏信息、被篡改、刪除或隱藏的情況，如果項目方提供的文件和資料存在不真實、不準確、缺乏信息、被篡改、刪除或隱藏的情況，或者對上述文件和資料的改動是在發布審計報告之後作出的，我方不承擔因反映情況與實際情況不一致引起的損失和不利影響 。

iv. 項目方知曉我方出具的審計報告系根據項目方提供的文件和資料、依靠我方現掌握的技術而作出的。但由於任何機構均存在技術的局限性，我方作出的審計報告仍存在無法完整檢測出全部風險的可能性。我方審計團隊鼓勵項目的開發團隊以及任何相關利益方對項目進行後續的測試及審計。

v. 項目方保證其委托我方提供審計或測試服務的項目合法、合規，且不違反適用法律。審計報告僅用於項目方參考，審計報告的內容、獲取方式、使用以及任何其所涉及的服務或資源都不能作為任何形式的投資、稅務、法律、監管及建議等的依據，我方不因此承擔相關責任。在未經我方書面同意之前，項目方不得將審計報告的全部或部分內容以任何形式提及、引用、展示或發送給任何第三方，否則由此產生的任何損失和責任由項目方自行承擔。我方對任何人依賴審計報告或將之用於任何目的概不承擔責任 。

vi. 本審計報告不涉及合約的編譯器及任何超出智能合約編程語言的領域，所審計的智能合約因引用鏈下信息或資源所導致的風險及責任，由項目方自行承擔 。

vii. 不可抗力。不可抗力是指雙方在訂立合同時不能預見、對其發生和後果不能避免且不能克服的事件，包括但不限於戰爭、臺風、水災、火災、地震、潮汐、雷電、天災、罷工、核爆炸、流行病等自然災害和法律、法規和政策變更及政府行為等其它不可預見，對其發生和後果不能防止或避免的事件，且該事件妨礙、影響或延誤任何一方根據合同履行其全部或部分義務。

viii. 如果有一方認為不可抗力發生影響履行本協議義務，應迅速通知另一方，按事件對履約影響的程度，由雙方協商決定是否終止合同或部分免除履約的責任，或者延期履約。

ix. 當不可抗力發生時，任何一方都不能被視作違約或不履行本協議義務。在事件前存在的經濟上的責任，不應受到影響，項目方應對我方已完成工作做出支付。

2022年6月6日

審計通過.

日期 2022年6月6日

審計 歐科雲鏈

本次審計的目的是為了審閱OKC 價格預言機項目基於Solidity語言編寫的餵價、獲取價格等協議，研究其設計、架構，發現潛在的安全隱患，並試圖找到可能存在的漏洞。