VITAL BLOCK

Security Assessment

ARBITRUM EXCHANCE

Vital Block Verified on March 17th, 2023

@Vital-Block

@VB_Audit

info@vitalblock.org

www.vitalblock.org

PREPARED FOR:

ARBITRUM
EXCHANCE

VITALBLOCK SMART CONTRACT AUDIT

INTRODUCTION

Auditing Firm	**VITAL BLOCK SECURITY**
Client Firm	**ABITRUM EXCHANGE**
Methodology	**Automated Analysis, Manual Code Review**
Language	**Solidity**
Contract's	ARXToken: 0xD5954c3084a1cCd70B4dA011E67760B8e78aeE84 DummyToken: 0x5DD7cB04Ed941F6919aB42519F13662323a16e24 (Used when initializing ARXPool) Masterchef: 0xeb51F3346626CBB79c1b839C83Bf008cFc713231 Router: 0x3E48298A5Fe88E4d62985DFf65Dee39a25914975 Factory: 0x1C6E968f2E6c9DEC61DB874E28589fd5CE3E1f2c ARXPool: 0x20B09797128c189A940fAE69af6fC6D002F576B7 ArbiFlexPool: 0x4c56a8A55b946f4Eef20C1cfe661f18f7Ff1BCBD SmartChefFactory: 0x086CdB9aA631270F4d14E9360735eeE86c6505e9 Earn WBTC: 0x907E5d334F27a769EF779358089fE5fdAA6cf2Bb Earn WETH: 0x75Bca51be93E97FF7D3198506f368b472730265a Earn USDC: 0x466f4380327cD948572AE0C98f2E04930ce05767
Blockchain	**ARBITRUM**
Centralization	**Active ownership**
Website	**https://arbidex.fi**
Discord	**https://discord.gg/arbitrumexchange**
Twitter	**https://twitter.com/Arbidex_fi**
GitHub	**https://github.com/fractalityy/ArbiDex/tree/master**
Prelim Report Date	**MARCH 16, 2023**
Final Report Date	**MARCH 17, 2023**

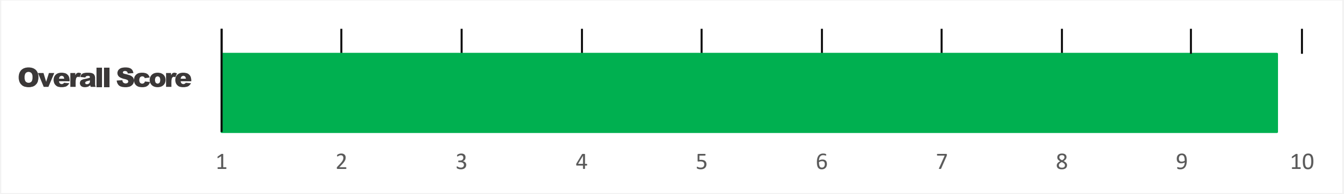
ⓘ **Verify the authenticity of this report on our GitHub Repo: https://www.github.com/vital-block**

EXECUTIVE SUMMARY

Vital Block Security has performed the automated and manual analysis of the Sol code. The code was reviewed for common contract vulnerabilities and centralized exploits. Here's a quick audit summary:

Status	Critical ! 🔴	Major " 🟠	Medium # 🟡	Minor $ 🟢	Unknown % 🟤
Open	0	0	0	2	0
Acknowledged	0	0	1	5	0
Resolved	0	0	0	0	0
Noteworthy onlyOwner Privileges	Set Taxes and Ratios, Airdrop, Set Protection Settings, Set Reward Properties, Set Reflector Settings, Set Swap Settings, Set Pair and Router				

ARBITRUM EXCHANGE Smart contract has achieved the following score: 98.5



ℹ **Please note that smart contracts deployed on blockchains aren't resistant to exploits, vulnerabilities and/or hacks. Blockchain and cryptography assets utilize new and emerging technologies. These technologies present a high level of ongoing risks. For a detailed understanding of risk severity, source code vulnerability, and audit limitations, kindly review the audit report thoroughly.**

ℹ **Please note that centralization privileges regardless of their inherited risk status - constitute an elevated impact on smart contract safety and security.**

TABLE OF CONTENTS

SCOPE OF WORK

Vital Block was consulted by **ABITRUM EXCHANGE** to conduct the smart contract audit of its. Sol source code. The audit scope of work is strictly limited to mentioned .SOL file only:

- **ARBDEX TOKEN.Sol**

ℹ External contracts and/or interfaces dependencies are not checked due to being out of scope.

Verify audited contract's contract address and deployed link below:

Public Contract Link	
ARX: 0xD5954c3084a1cCd70B4dA011E67760B8e78aeE84	
Contract Name	**ArbiDex Token**
Token Symbol	ARX
Total Supply	171,720
Decimals	18

AUDIT METHODOLOGY

Smart contract audits are conducted using a set of standards and procedures. Mutual collaboration is essential to performing an effective smart contract audit. Here's a brief overview of Vital Block auditing process and methodology:

CONNECT

- **The onboarding team gathers source codes, and specifications to make sure we understand the size, and scope of the smart contract audit.**

AUDIT

- **Automated analysis is performed to identify common contract vulnerabilities. We may use the following third-party frameworks and dependencies to perform the automated analysis:**
    - **Remix IDE Developer Tool**
    - **Open Zeppelin Code Analyzer**
    - **SWC Vulnerabilities Registry**
    - **DEX Dependencies, e.g., Pancakeswap, Uniswap**
- **Simulations are performed to identify centralized exploits causing contract and/or trade locks.**
- **A manual line-by-line analysis is performed to identify contract issues and centralized privileges. We may inspect below mentioned common contract vulnerabilities, and centralized exploits:**

Centralized Exploits	○ **Token Supply Manipulation** ○ **Access Control and Authorization** ○ **Assets Manipulation** ○ **Ownership Control** ○ **Liquidity Access** ○ **Stop and Pause Trading** ○ **Ownable Library Verification**

Common Contract Vulnerabilities

- ○ Integer Overflow
- ○ Lack of Arbitrary limits
- ○ Incorrect Inheritance Order
- ○ Typographical Errors
- ○ Requirement Violation
- ○ Gas Optimization
- ○ Coding Style Violations
- ○ Re-entrancy
- ○ Third-Party Dependencies
- ○ Potential Sandwich Attacks
- ○ Irrelevant Codes
- ○ Divide before multiply
- ○ Conformance to Solidity Naming Guides
- ○ Compiler Specific Warnings
- ○ Language Specific Warnings

REPORT

- ○ **The auditing team provides a preliminary report specifying all the checks which have been performed and the findings thereof.**
- ○ **The client's development team reviews the report and makes amendments to the codes.**
- ○ **The auditing team provides the final comprehensive report with open and unresolved issues.**

PUBLISH

- ○ **The client may use the audit report internally or disclose it publicly.**

ℹ **It is important to note that there is no pass or fail in the audit, it is recommended to view the audit**

as an unbiased assessment of the safety of solidity codes.

RISK CATEGORIES

Smart contracts are generally designed to hold, approve, and transfer tokens. This makes them very tempting attack targets. A successful external attack may allow the external attacker to directly exploit. A successful centralization-related exploit may allow the privileged role to directly exploit. All risks which are identified in the audit report are categorized here for the reader to review:

Risk Type	Definition
Critical ! 🔴	**These risks could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.**
Major " 🟠	**These risks are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which canlead to high-risk severity.**
Medium # 🟡	**These risks should be fixed, as they carry an inherent risk of future exploits, and hacks which may or may not impact the smart contract execution. Low-risk re-entrancy-related vulnerabilities should be fixed to deterexploits.**
Minor $ 🟢	**These risks do not pose a considerable risk to the contract or those who interact with it. They are code-style violations and deviations from standard practices. They should be highlighted and fixed nonetheless.**
Unknown % 🟤	**These risks pose uncertain severity to the contract or those who interact with it. They should be fixed immediately to mitigate the riskuncertainty.**

All statuses which are identified in the audit report are categorized here for the reader toreview:

Status Type	Definition
Open	**Risks are open.**
Acknowledged	**Risks are acknowledged, but not fixed.**
Resolved	**Risks are acknowledged and fixed.**

CENTRALIZED PRIVILEGES

Centralization risk is the most common cause of cryptography asset loss. When a smart contract has a privileged role, the risk related to centralization iselevated.

There are some well-intended reasons haveprivileged roles, such as:

- **Privileged roles can be granted the power to** pause()**the contract in case of an external attack.**
- **Privileged roles can use functions like**, include(), **and** exclude() **to add or remove wallets from fees, swap checks, and transaction limits. This is useful to run a presale and to list on an exchange.**

Authorizing privileged roles to externally-owned-account (EOA) is dangerous. Lately, centralization-related losses are increasing in frequency and magnitude.

- **The client can lower centralization-related risks by implementing below mentioned practices:**
- **Privileged role's private key must be carefully secured to avoid any potential hack.**
- **Privileged role should be shared by multi-signature (multi-sig) wallets.**
- **Authorized privilege can be locked in a contract, user voting, or community DAO can be introduced to unlock the privilege.**
- **Renouncing the contract ownership, and privileged roles.**
- **Remove functions with elevated centralization risk.**

ℹ **Understand the project's initial asset distribution. Assets in the liquidity pair should be locked. Assets outside the liquidity pair should be locked with a release schedule.**

AUDIT SCOPE

ARBITRUM EXCHANGE

ID	Repo	Comment	File	SHM321 Checksum
ABY	contracts/fractality/Arbidex	cC512486	ARXFlexiblepool.sol	6788099YIRHVSK853PKFMGHEF44309200KDHFCBUGIJN
ABI	contracts/fractality/Arbidex	cC512486	ARXPool.sol	347520JHDB7549H22H3BVDIOETYUHF009JBIKBDI33BJ4
ABW	contracts/fractality/Arbidex	cC512486	ARXToken.sol	1988Y73HUGFDINN353840NFMTEJER73649RGFIMDIDH
ABG	contracts/fractality/Arbidex	cC512486	MasterChefV2.sol	4438648TEOHBF6378309EHROECNEPOEJDNETE8EYEU3
ABL	contracts/fractality/Arbidex	cC512486	Factory.sol	66390028765RVNKDBYFTGW553T2KOEHIUUJJIJE
ABA	contracts/fractality/Arbidex	cC512486	Router.sol	09825539BDYG543DVNKOMIKEBYR JUFHHFHJFIE333222
ABJ	contracts/fractality/Arbidex	cC512486	ArxTokenV2ABI,json	8654RJVT3DWI865YK26437903JJDGGDHGWY6E
ABE	contracts/fractality/Arbidex	cC512486	MasterChefv2.sol	7763888636TGYGFFTFHBETT66TFTCTVYBHBYT
ABP	contracts/fractality/Arbidex	cC512486	Zapper.sol	88530486494YRHFTEICBGEIEGWTWYWUHEJEHEIE33U3
ABM	contracts/fractality/Arbidex	cC512486	ArxTokenV2ABI.json	1209873KHJLKJNFJHGE98763990029774BCUHHDUU239
ABV	contracts/fractality/Arbidex	cC512486	ARXPoolABI.json	23456UGFYUHE98756EFHJHE7654ESDFGHGERTYUJ3897
ABQ	contracts/fractality/Arbidex	cC512486	Presale.sol	37889UHBIONEO7TYRDFGVBN5678939IJWSFVDYUHDCI
ABS	contracts/fractality/Arbidex	cC512486	ArbDexPairABI.json	678903098TFHJKFCPOIUGFGHJKE9865ERGBEIVBHE8767
ABR	contracts/fractality/Arbidex	cC512480	SmartChefInitializableABI.json	98765SDFGBNFCOI56789UIYHGGHEJDIUYTRDCVBN3459

AUTOMATED ANALYSIS

Symbol	Definition
🛑	**Function modifies state**
💵	**Function is payable**
🔒	**Function is internal**
🔐	**Function is private**
❗	**Function is important**

```
| **ARBDEX TOKEN** | Interface |  |||

| └ | totalSupply | External ▯ |  ❗  |NO▯ |

| └ | decimals | External ▯ |  ❗  |NO▯ |

| └ | symbol | External ▯ |  ❗  |NO▯ |

| └ | name | External ▯ |  ❗  |NO▯ |

| └ | getOwner | External ▯ |  |NO▯ |

| └ | balanceOf | External ▯ |  ❗  |NO▯ |

| └ | transfer | External ▯ | "  ❗  🛑  |NO▯ |

| └ | allowance | External ▯ |  ❗  |NO▯ |

| └ | approve | External ▯ | "  ❗  🛑  |NO▯ |

| └ | transferFrom | External ▯ | "  |NO▯ |

||||||

| **IFactoryV2** | Interface |  |||

| └ | getPair | External ▯ |  |NO▯ |

| └ | createPair | External ▯ | "  |NO▯ |

||||||

| **IV2Pair** | Interface |  |||

| └ | factory | External ▯ |  |NO▯ |

| └ | getReserves | External ▯ |  |NO▯ |

| └ | sync | External ▯ | "  |NO▯ |
||||||
| **IRouter01** | Interface |  |||
| └ | factory | External ▯ |  |NO▯ |
| └ | ETH | External ▯ |  |NO▯ |
| └ | addLiquidityETH | External ▯ |  #  |NO▯ |
| └ | addLiquidity | External ▯ | "  |NO▯ |
| └ | swapExactAPTForTokens | External ▯ |  #  |NO▯ |
| └ | getAmountsOut | External ▯ |  |NO▯ |
| └ | getAmountsIn | External ▯ |  |NO▯ |
||||||
| **IRouter02** | Interface | IRouter01 |||
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ▯ | "  |NO▯ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ▯ |  #  |NO▯ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ▯ | " ! 🛑 |NO▯ |
| └ | swapExactTokensForTokens | External ▯ | "  |NO▯ |
||||||
| **Protections** | Interface |  |||
| └ | checkUser | External ▯ | " ! 🛑 |NO▯ |
| └ | setLaunch | External ▯ | " ! 🛑 |NO▯ |
| └ | setLpPair | External ▯ | " ! 🛑 |NO▯ |
| └| ARX | External ▯ | "  |NO▯ |
| └ | removeSniper | External ▯ | "  |NO▯ |
||||||
| **Cashier** | Interface |  |||
| └ | setRewardsProperties | External ▯ | "  |NO▯ |
| └ | tally | External ▯ | "  |NO▯ |
| └ | load | External ▯ |  #  |NO▯ |
| └ | cashout | External ▯ | "  |NO▯ |
| └ | giveMeWelfarePlease | External ▯ | "  |NO▯ |
| └ | getTotalDistributed | External ▯ |  |NO▯ |
| └ | getUserInfo | External ▯ |  |NO▯ |
| └ | getUserRealizedRewards | External ▯ |  |NO▯ |
| └ | getPendingRewards | External ▯ |              |NO▯ |

| └ | initialize | External ▯ | "                  |NO▯ |

| └ | getCurrentReward | External ▯ |              |NO▯ |

||||||

| **SOL** | Implementation | SafeMath |||

| └ | <Constructor> | Public ▯ |              #  |NO▯ |

| └ | transferOwner | External ▯ | "              | onlyOwner |

| └ | renounceOwnership | External ▯ | "             | NO!|

| └ | setOperator | Public ▯ | "                 |NO▯ |

| └ | renounceOriginalDeployer | External ▯ | "            |NO▯ |

| └ | <Receive Ether> | External ▯ |              #  |NO▯ |

| └ | totalSupply | External ▯ |                 |NO▯ |

| └ | decimals | External ▯ |                 |NO▯ |

| └ | symbol | External ▯ |                 |NO▯ |

| └ | name | External ▯ |                 |NO▯ |

| └ | getOwner | External ▯ |       !        |NO▯ |

| └ | balanceOf | Public ▯ |        !       |NO▯ |

| └ | allowance | External ▯ |      !         |NO▯ |

| └ | approve | External ▯ | " !      🛑       |NO▯ |

| └ | _approve | Internal $  | "                | |

| └ | approveContractContingency | Public ▯ | "     !      🛑     | onlyOwner |

| └ | transfer | External ▯ | "                 |NO▯ |

| └ | transferFrom | External ▯ | "                |NO▯ |

| └ | setNewRouter | External ▯ | "              | onlyOwner |

| └ | setLpPair | External ▯ | "                | onlyOwner |

| └ | setInitializers | External ▯ | "              | onlyOwner |

| └ | isExcludedFromFees | External ▯ |              |NO▯ |

| └ | isExcludedFromDividends | External ▯ |              |NO▯ |

| └ | isExcludedFromProtection | External ▯ |              |NO▯ |

| └ | setDividendExcluded        | Public ▯ | "            | onlyOwner |

| └ | setExcludedFromFees        | Public ▯ | "            | onlyOwner |
```

OPTIMIZATIONS | ARBITRUM EXCHANGE

ID	Title	Category	Status
STV	**Logarithm Refinement Optimization**	Gas Optimization	Acknowledged ●
SOP	**Checks Can Be Performed Earlier**	Gas Optimization	Acknowledged ●
SDP	**Unnecessary Use Of SafeMath**	Gas Optimization	Acknowledged ●
SWY	**Struct Optimization**	Gas Optimization	Acknowledged ●
SGT	**Unused State Variable**	Gas Optimization	Acknowledged ●

General Detectors

! Floating Pragma

This contract may not function as expected due to inconsistent solidity compiler versions being specified

⚠ Attention Required

! Low Level Calls

This contract uses low-level calls, which may be unsafe.

⚠ Attention Required

! Numeric Notation Best Practices

The numeric notation used in this contract is unconventional, possibly worsening the reading/debugging experience

⚠ Attention Required

- ✓ No compiler version inconsistencies found
- ✓ No unchecked call responses found
- ✓ No vulnerable self-destruct functions found
- ✓ No assertion vulnerabilities found
- ✓ No old solidity code found
- ✓ No external delegated calls found
- ✓ No external call dependency found
- ✓ No vulnerable authentication calls found
- ✓ No invalid character typos found
- ✓ No RTL characters found
- ✓ No dead code found
- ✓ No risky data allocation found
- ✓ No uninitialized state variables found
- ✓ No uninitialized storage variables found
- ✓ No vulnerable initialization functions found
- ✓ No risky data handling found
- ✓ No number accuracy bug found
- ✓ No out-of-range number vulnerability found
- ✓ No map data deletion vulnerabilities found
- ✓ No tautologies or contradictions found
- ✓ No faulty true/false values found
- ✓ No innacurate divisions found
- ✓ No redundant constructor calls found
- ✓ No vulnerable transfers found
- ✓ No vulnerable return values found
- ✓ No uninitialized local variables found
- ✓ No default function responses found
- ✓ No missing arithmetic events found
- ✓ No missing access control events found
- ✓ No redundant true/false comparisons found
- ✓ No state variables vulnerable through function calls found
- ✓ No buggy low-level calls found
- ✓ No expensive loops found
- ✓ No bad numeric notation practices found
- ✓ No missing constant declarations found
- ✓ No missing external function declarations found
- ✓ No vulnerable payable functions found
- ✓ No vulnerable message values found

Vulnerability Scan

REENTRANCY

✓ No reentrancy risk found

Severity	Major
Confidence Parameter	Certain

Vulnerability Description

✗ **Mintable:** More amount of this token can be minted by a private wallet or contract. **(This is Essentially normal for most contracts)**

Scanning Line:

```
function _functionCallWithValue(
        address target,
        bytes memory data,
        uint256 weiValue,
        string memory errorMessage
    ) private returns (bytes memory) {
        require(isContract(target), 'Address: call to non-contract');

        // solhint-disable-next-line avoid-low-level-calls
        (bool success, bytes memory returndata) = target.call{value: 
weiValue}(data);
        if (success) {
            return returndata;
        } else {
            // Look for revert reason and bubble it up if present
            if (returndata.length > 0) {
                // The easiest way to bubble the revert reason is using memory via 
assembly

                // solhint-disable-next-line no-inline-assembly
                assembly {
                    let returndata_size := mload(returndata)
                    revert(add(32, returndata), returndata_size)
                }
            } else {
```

Repository:

https://github.com/fractalityy/ArbiDex/tree/master

All Audited Files

- ARX Token.sol
- Dummy Token.sol
- Masterchef.sol
- Router.sol
- Factory.sol
- ARXPool.sol
- ArbiFlexPool.spl
- SmartChefFactory.sol
- Earn WBTC.sol
- Earn WETH.sol
- Earn USDC.sol

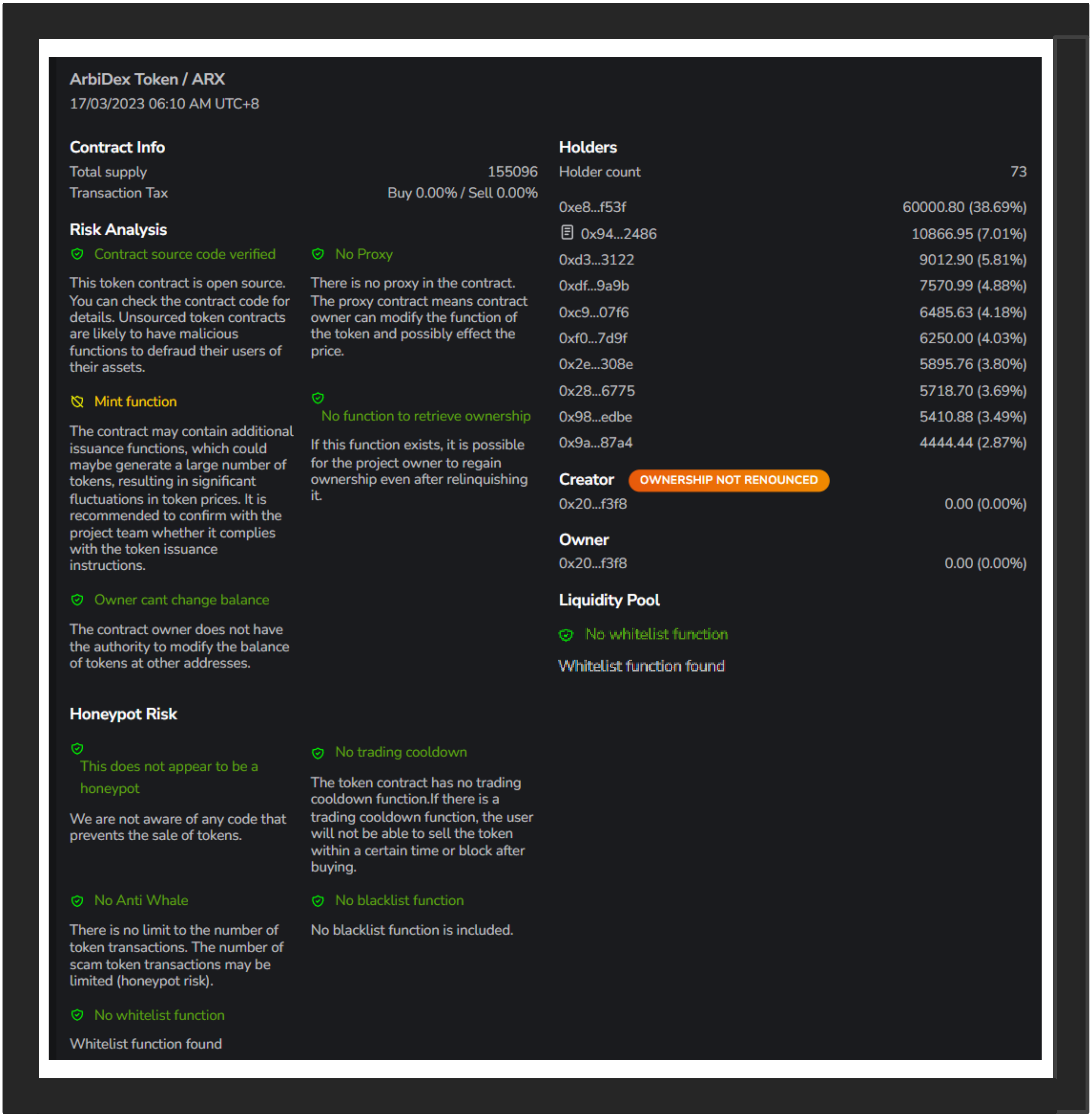
Contract Creator 0x2084e8ecdca037e4751a8ead62ebd324425ff3f8

Creator Tnx Hash 0xfbddc1ad558290ae471dc975143bdc4ee3681eb2611e5943f45e5d9c45f0ec14

Contracts:

```
Contract:
ARX Token: 0xD5954c3084a1cCd70B4dA011E67760B8e78aeE84
Dummy Token: 0x5DD7cB04Ed941F6919aB42519F13662323a16e24
(Used when initializing ARXPool)
Masterchef: 0xeb51F3346626CBB79c1b839C83Bf008cFc713231
Router: 0x3E48298A5Fe88E4d62985DFf65Dee39a25914975
Factory: 0x1C6E968f2E6c9DEC61DB874E28589fd5CE3E1f2c
ARXPool: 0x20B09797128c189A940fAE69af6fC6D002F576B7
ArbiFlexPool: 0x4c56a8A55b946f4Eef20C1cfe661f18f7Ff1BCBD
SmartChefFactory: 0x086CdB9aA631270F4d14E9360735eeE86c6505e9
Earn WBTC: 0x907E5d334F27a769EF779358089fE5fdAA6cf2Bb
Earn WETH: 0x75Bca51be93E97FF7D3198506f368b472730265a
Earn USDC: 0x466f4380327cD948572AE0C98f2E04930ce05767
```

Vulnerability Run check



ArbiDex Token / ARX
17/03/2023 06:10 AM UTC+8

Contract Info

Total supply	155096
Transaction Tax	Buy 0.00% / Sell 0.00%

Risk Analysis

✓ Contract source code verified

This token contract is open source. You can check the contract code for details. Unsourced token contracts are likely to have malicious functions to defraud their users of their assets.

Mint function

The contract may contain additional issuance functions, which could maybe generate a large number of tokens, resulting in significant fluctuations in token prices. It is recommended to confirm with the project team whether it complies with the token issuance instructions.

✓ Owner cant change balance

The contract owner does not have the authority to modify the balance of tokens at other addresses.

✓ No Proxy

There is no proxy in the contract. The proxy contract means contract owner can modify the function of the token and possibly effect the price.

✓ No function to retrieve ownership

If this function exists, it is possible for the project owner to regain ownership even after relinquishing it.

Honeypot Risk

✓ This does not appear to be a honeypot

We are not aware of any code that prevents the sale of tokens.

✓ No Anti Whale

There is no limit to the number of token transactions. The number of scam token transactions may be limited (honeypot risk).

✓ No whitelist function

Whitelist function found

✓ No trading cooldown

The token contract has no trading cooldown function.If there is a trading cooldown function, the user will not be able to sell the token within a certain time or block after buying.

✓ No blacklist function

No blacklist function is included.

Holders

Holder count	73
0xe8...f53f	60000.80 (38.69%)
0x94...2486	10866.95 (7.01%)
0xd3...3122	9012.90 (5.81%)
0xdf...9a9b	7570.99 (4.88%)
0xc9...07f6	6485.63 (4.18%)
0xf0...7d9f	6250.00 (4.03%)
0x2e...308e	5895.76 (3.80%)
0x28...6775	5718.70 (3.69%)
0x98...edbe	5410.88 (3.49%)
0x9a...87a4	4444.44 (2.87%)

Creator OWNERSHIP NOT RENOUNCED

0x20...f3f8	0.00 (0.00%)

Owner

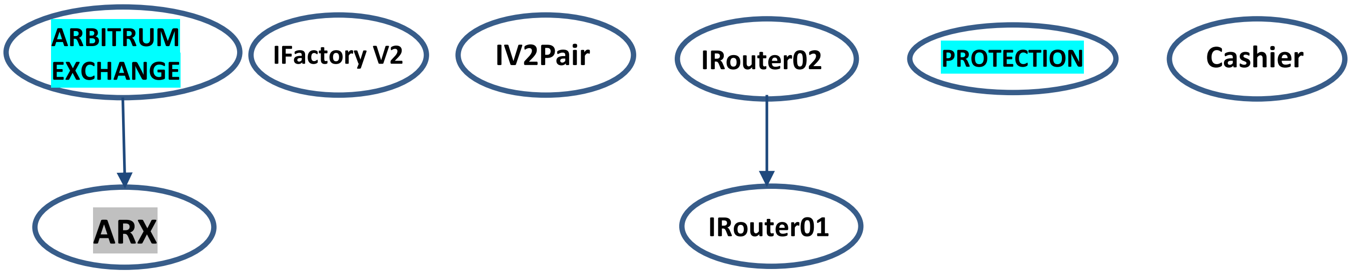
0x20...f3f8	0.00 (0.00%)

Liquidity Pool

✓ No whitelist function

Whitelist function found

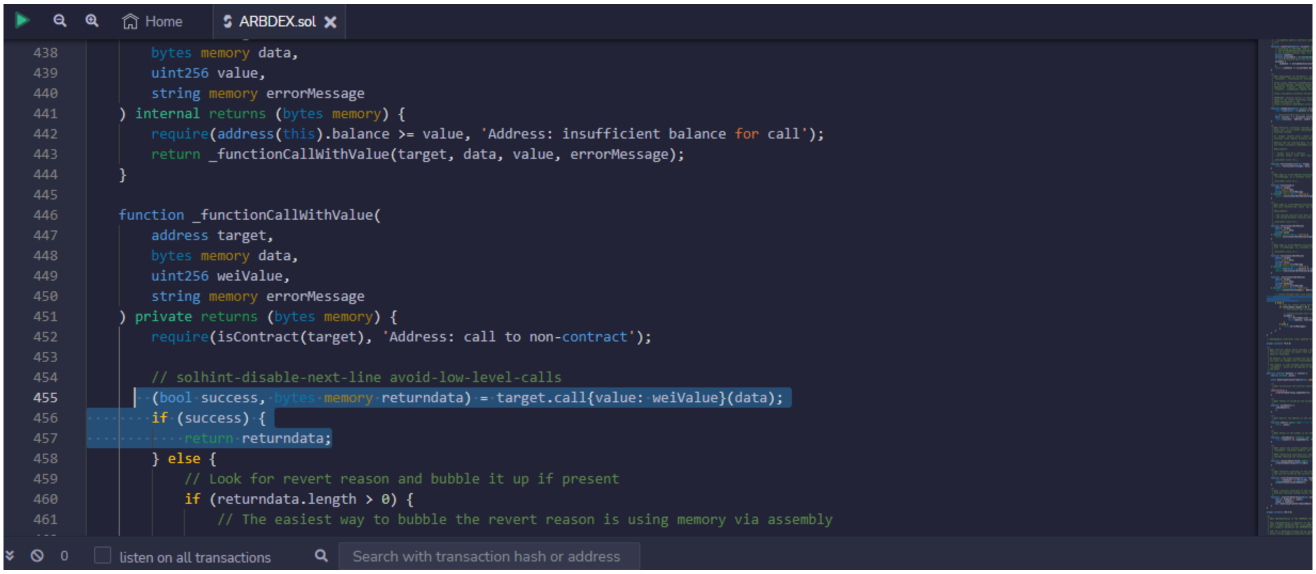
INHERITANCE GRAPH



Identifier	Definition	Severity
CEN-12	Centralization privileges of ARBITRUM EXCHANGE	Medium # 🟡

Vulnerability 0 : No important security issue detected.

Threat level: Low



```
        bytes memory data,
        uint256 value,
        string memory errorMessage
    ) internal returns (bytes memory) {
        require(address(this).balance >= value, 'Address: insufficient balance for call');
        return _functionCallWithValue(target, data, value, errorMessage);
    }

    function _functionCallWithValue(
        address target,
        bytes memory data,
        uint256 weiValue,
        string memory errorMessage
    ) private returns (bytes memory) {
        require(isContract(target), 'Address: call to non-contract');

        // solhint-disable-next-line avoid-low-level-calls
        (bool success, bytes memory returndata) = target.call{value: weiValue}(data);
        if (success) {
            return returndata;
        } else {
            // Look for revert reason and bubble it up if present
            if (returndata.length > 0) {
                // The easiest way to bubble the revert reason is using memory via assembly
```

0
listen on all transactions
Search with transaction hash or address

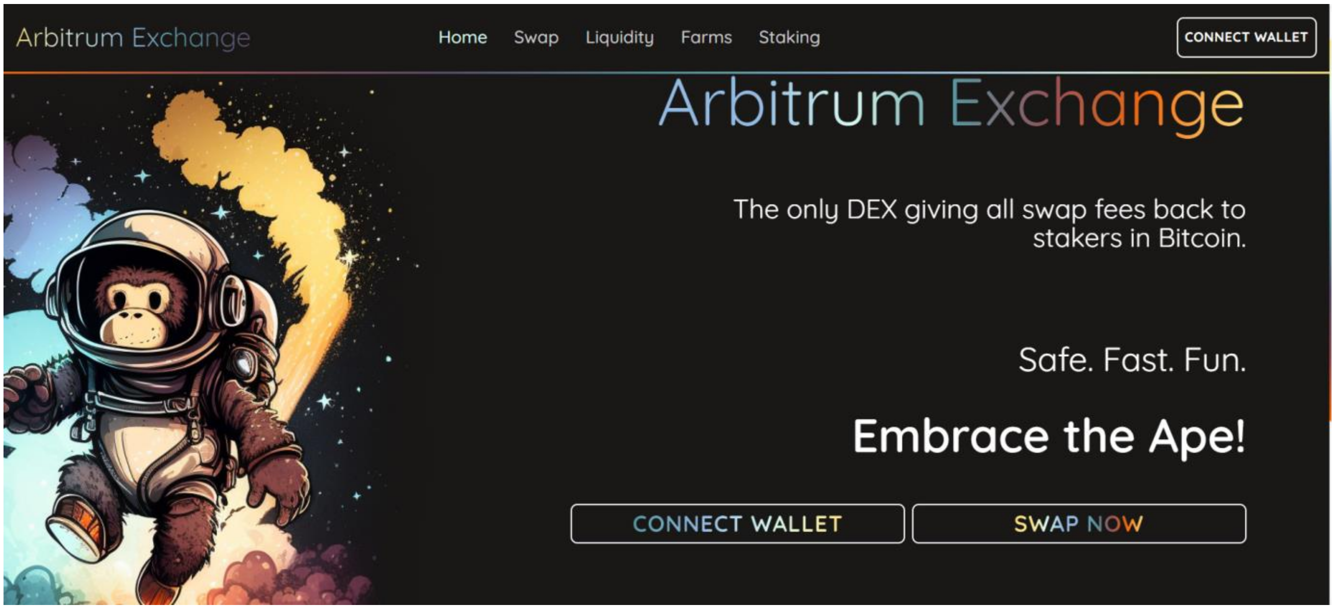
MANUAL REVIEW

ARBITRUM EXCHANGE: ARBDEX IS THE MOST SECURE COMMUNITY-DRIVEN REWARDING DEX ON ARBITRUM NETWORK.

TOKEN NAME: ARBDEX TOKEN
Ticker: **ARX**

Chain/Standard: ARBITRUM BLOCKCHAIN

The ARBITRUM EXCHANGE Platform Is Launched On Arbitrum



Arbitrum Exchange

The only DEX giving all swap fees back to stakers in Bitcoin.

Safe. Fast. Fun.

Embrace the Ape!

ISSUES CHECKING STATUS

	Issue Description	Checking Status
1.	Compiler errors.	PASSED
2.	Race Conditions and reentrancy. Cross-Function Race Conditions.	PASSED
3.	Possible Delay In Data Delivery.	PASSED
4.	Oracle calls.	PASSED
5.	Front Running.	PASSED
6.	Sol Dependency.	PASSED
7.	Integer Overflow And Underflow.	PASSED
8.	DoS with Revert.	PASSED
9.	Dos With Block Gas Limit.	PASSED
10.	Methods execution permissions.	PASSED
11.	Economy Model of the contract.	PASSED
12.	The Impact Of Exchange Rate On the solidity Logic.	PASSED
13.	Private use data leaks.	PASSED
14.	Malicious Event log.	PASSED
15.	Scoping and Declarations.	PASSED
16.	Uninitialized storage pointers.	PASSED
17.	Arithmetic accuracy.	PASSED
18.	Design Logic.	PASSED
19.	Cross-Function race Conditions	PASSED
20.	Save Upon solidity contract Implementation and Usage.	PASSED
21.	Fallback Function Security	PASSED

AUDIT RESULT

PASSED

Identifier	Definition	Severity
CEN-02	Initial asset distribution	Minor ●

All of the initially minted assets are sent to the contract deployer when deploying the contract. This is Normal for most deployer and/or contract owner .

```
/
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value,
        string memory errorMessage
    ) internal returns (bytes memory) {
        require(address(this).balance >= value, 'Address: insufficient balance for call');
        return _functionCallWithValue(target, data, value, errorMessage);
```

RECOMMENDATION

Project stakeholders should be consulted during the initial asset distribution process.

RECOMMENDATION

Deployer and/or contract owner private keys are secured carefully.

Please refer to PAGE-09 CENTRALIZED PRIVILEGES for a detailed understanding.

ALLEVIATION

The ARBITRUM EXCHANGE project team understands the centralization risk. Some functions are provided privileged access to ensure a good runtime behavior in the project

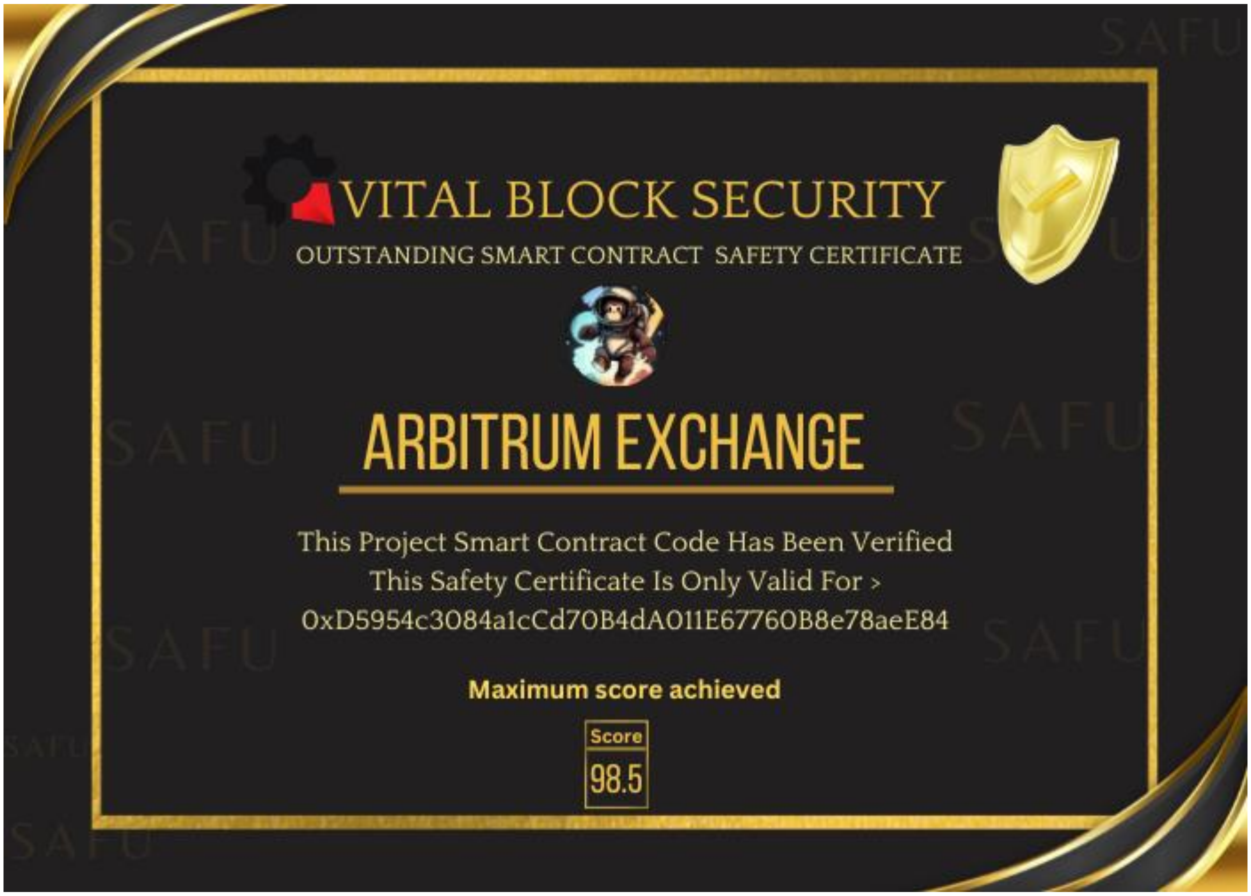
Identifier	Definition	Severity
COD-10	Third Party Dependencies	Minor 🟢

Smart contract is interacting with third party protocols e.g., Pancakeswap router, cashier contract, protections contract. The scope of the audit treats third party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised, and exploited. Moreover, upgrades in third parties can create severe impacts, e.g., increased transactional fees, deprecation of previous routers, etc.

RECOMMENDATION

Inspect and validate third party dependencies regularly, and mitigate severe impacts whenever necessary.

CERTIFICATE BY VITAL BLOCK SECURITY



VITAL BLOCK SECURITY

OUTSTANDING SMART CONTRACT SAFETY CERTIFICATE

ARBITRUM EXCHANGE

This Project Smart Contract Code Has Been Verified
This Safety Certificate Is Only Valid For >
0xD5954c3084a1cCd70B4dA011E67760B8e78aeE84

Maximum score achieved

Score
98.5

DISCLAIMERS

Vital Block provides the easy-to-understand audit of Solidity, Move and Raw source codes (commonly known as smart contracts).

The smart contract for this particular audit was analyzed for common contract vulnerabilities, and centralization exploits. This audit report makes no statements or warranties on the security of the code. This audit report does not provide any warranty or guarantee regarding the absolute bug-free nature of the smart contract analyzed, nor do they provide any indication of the client's business, business model or legal compliance. This audit report does not extend to the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. Cryptographic tokens are emergent technologies, they carry high levels of technical risks and uncertainty. 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. This audit report could include false positives, false negatives, and other unpredictable results.

CONFIDENTIALITY

This report is subject to the terms and conditions (including without limitations, description of services, confidentiality, disclaimer and limitation of liability) outlined in the scope of the audit provided to the client. This report should not be transmitted, disclosed, referred to, or relied upon by any individual for any purpose without InterFi Network's prior written consent.

NO FINANCIAL ADVICE

This audit report does not indicate the endorsement of any particular project or team, nor guarantees its security. No third party should rely on the reports in any way, including to make any decisions to buy or sell a product, service or any other asset. The information provided in this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. This audit report should not be used in any way to make decisions around investment or involvement. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

FOR AVOIDANCE OF DOUBT, SERVICES, INCLUDING ANY ASSOCIATED AUDIT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

TECHNICAL DISCLAIMER

ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, VITAL BLOCK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO SERVICES, AUDIT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, VITAL BLOCK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, VITAL BLOCK MAKES NO WARRANTY OF ANY KIND THAT ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE CLIENT'S OR ANY OTHER INDIVIDUAL'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE.

TIMELINESS OF CONTENT

The content contained in this audit report is subject to change without any prior notice. Vital Block does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following the publication.

LINKS TO OTHERWEBSITES

This audit report provides, through hypertext or other computer links, access to websites and social accounts operated by individuals other than Vital Block. Such hyperlinks are provided for your reference and convenience only and are the exclusive responsibility of such websites and social accounts owners. You agree that Vital block Security is not responsible for the content or operation of such websites and social accounts and that Vital Block shall have no liability to you or any other person or entity for the use of third-party websites and social accounts. You are solely responsible for determining the extent to which you may use any content at any other websites and social accounts to which you link from the report.

ABOUT VITAL BLOCK

Vital Block provides intelligent blockchain Security Solutions. We provide solidity and Raw Code Review, testing, and auditing services. We have Partnered with 15+ Crypto Launchpads, audited 50+ smart contracts, and analyzed 200,000+ code lines. We have worked on major public blockchains e.g., Ethereum, Binance, Cronos, Doge, Polygon, Avalanche, Metis, Fantom, Bitcoin Cash, Aptos, Oasis, etc.

Vital Block is Dedicated to Making Defi & Web3 A Safer Place. We are Powered by Security engineers, developers, UI experts, and blockchain enthusiasts. Our team currently consists of 5 core members, and 4+ casual contributors.

Website: https://Vitalblock.org

Email: info@vitalblock.org

GitHub: https://github.com/vital-block

Telegram (Engineering): https://t.me/vital_block

Telegram (Onboarding): https://t.me/vitalblock_cmo vital-block

info@vitalblock.org

www.Vitalblock.org

VITALBLOCK
SMART CONTRACT AUDIT

Vital Block Dedicated to securing Public and Private Blockchain Ecosystem