VITAL BLOCK

SMART CONTRACT AUDIT

@Vital-Block

@VB_Audit

info@vitalblock.org

www.vitalblock.org

PREPARED FOR:

ARCADEUM

INTRODUCTION

Auditing Company	**VITAL BLOCK SECURITY**
Client Project	**ARCADEUM**
Methodology	**Automated Analysis, Manual Code Review**
Language	**Solidity**
License	**MIT**
Contract Address	**0x7F465507f058e17Ad21623927a120ac05CA32741**
Network	**ARBITRUM CHAIN**
Token Type	**ERC20**
Website	**https://www.arcadeum.io**
Telegram	**https://t.me/arcadeum**
Twitter	**https://twitter.com/arcadeum_io**
GitHub	**https://github.com/arcadeum**
Prelim Report Date	**February 19, 2023**
Final Report Date	**February 20, 2023**

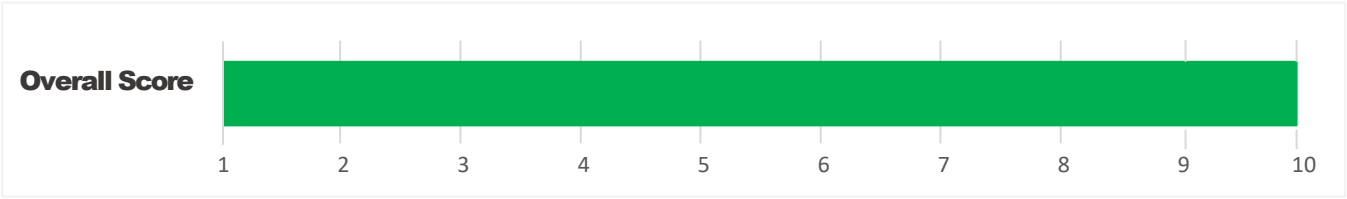
Verify the authenticity of this report on our GitHub Repo: https://www.github.com/vital-block

EXECUTIVE SUMMARY

Vital Block 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	1	1	2	0
Resolved	0	0	0	0	0
Noteworty OnlyOwner Privileges	Set Taxes and Ratios, Airdrop, Set Protection Settings, Set Reward Properties, Set Reflector Settings, Set Swap Settings, Set Pair and Router				

ARCADEUM Smart contract has achieved the following score: 99.7.



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.**

SCOPE OF WORK

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

- **ARCADEUM.Sol**

ℹ️ **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.	
0x7F465507f058e17Ad21623927a120ac05CA32741	
Contract Name	**ARCADEUM**
Token Symbol	**ARC**
Decimals	**18**
Blockchain	**Arbitrum Network**

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 can lead 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 deter exploits.
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 risk uncertainty.

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

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 is elevated.

There are some well-intended reasons have privileged 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.**

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.

Summary

Audit Report prepared by Vital Block Security covering the Arcadeum token sale smart contracts (and their associated components).

Process and Delivery

Three (7) independent Solidified experts performed an unbiased and isolated audit of the code
below. The debrief took place on February 18th, 2023.
This report constitutes of no amendment to the original audit report covering the Entire code
Fully audited on February 19th 2023.

Audited Files

The following contracts were covered during the audit:

contracts

- **ALP-** 0x875Ce69F87beB1d9B0c407597d4057Fa91D364A7
- **ARC-** 0x7F465507f058e17Ad21623927a120ac05CA32741
- **sARC-** 0xF97307D029643C7551c1e0eEfAB51206076735d3
- **sARCfees-** 0xD6Af60C6BC19C2f13558ebD5C779CA75f6Ccfdd2

contracts

- **xARC-** 0x1055f9956824c333AB52b6a87Cc1feec8797e473
- **xARCFees-** 0x5a71f02adDf8dB3935B898Ee27EB313565A7E632
- **esARC-** 0xf740366CB77094fF0CB51770d0eAFF58bF1697be
- **esARCFees-** 0x8f6c8a1077ee3C4dA03852235e36223Cc50167e8
- **Console-** 0x773Aa71471e24b027c981d9bD5e6B81a82f7b81d
- **Player-** 0xB8F3C0b80D0dCE61cE674ee0e935b96C4cad25B8
- **House-** 0x6e433358023eC537172af6Ec6cd6276613d17B31

- **GameRoulette-** 0xAdC8bD9Ef156BBa378955E11A5cf3de25039546e
- **GameDice-** 0xC664b8D7E86C48C0162090B545bc49DC9395c50b
- **GameWheel-** 0xa79EAF9F4ec3e8db9150501f8Ba7dDB5b467880F
- **GameSlide-** 0x6b29f1958f2A214f8C44e10C9928db66432201bB
- **GameLimbo-** 0x01CaCe27d694C278DE190F8B626c4bB2d69a245F
- **GameRockPaperScissors-** 0x0B82A9b7659Bfa3De5fDB320e55E273051CaE6e9
- **GameCoinFlip-** 0xA22E051692449Af1E1C21ba17016ac24349C5aa8
- **RNG-** 0x13F12D77F04Ed4063f7709cE26aD44CE3d2a07d1
  - **VestingContractA-** 0xEAce9962d33f5D5738357FBFBe2ae6fe1C8Bd32A
- **VestingContractB-** 0x16ca198Af4a5b29866dfFb86CbA80d04E5CA6A6C

Supplied in the following source code repositories

https://arcadeum.gitbook.io/arcadeum/contracts

Audit Report for ARCADEUM PROJECT - February 2023

Notes:

The audit was based on commit numbers 0afb5d17e8c49d0ce10ba364gtt557477 and 28717430a3574b8ee1f0d2f880f6c0y54443309

Last commit number: 121bda79ce9a27c3033a263c483ac8734aa2b103

Intended Behavior

The smart contract implements a token trade based on a modified version of the Balancer liquidity pool smart contract. In order to function as a token sale, the following modifications have been applied:

- Only a controller can provide liquidity to the pool
- Swap fees will are set to 0
- Users can only swap one-way (intended for buying ARC with ETH)

All Contract Ownership

0xfc2ec93a918b44e30a163e7daeb02f62595407f3 Is The Owner Of All Contracts.

Summary

- **Owner is not able to change or set taxes (0% tax)**
- **Owner is not able to set a max amount for buys/sells/transfer**
- **Owner is not able to pause trades**
- **Owner is not able to mint new tokens**
- **Owner is not able to blacklist an arbitrary address**

Issues Found

Solidified found that the Arcadeum contracts contain no critical issue, no major
issues, and 1 minor issue, in addition to 3 informational notes.
We recommend all issues are amended, while the notes are up to the team's discretion, as it refers to best practices.

Vulnerability Run check

Contract Info

Total supply	9971173.330016416
Transaction Tax	Buy 0.00% / Sell 0.00%
Dex 1	UniswapV3

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.

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 mint function

Mint function is transparent or non-existent. Hidden mint functions may increase the amount of tokens in circulation and effect the price of the token.

No function to retrieve ownership

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

Owner cant change balance

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

Honeypot Risk

This does not appear to be a honeypot

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

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 Anti Whale

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

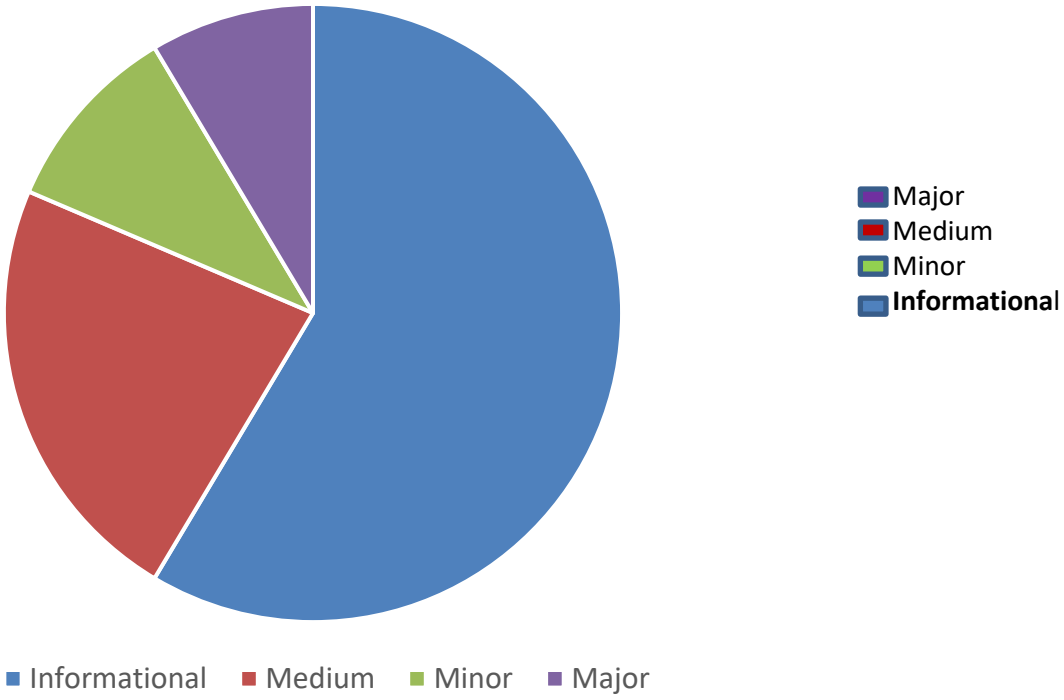
No blacklist function

No blacklist function is included.

No whitelist function

Whitelist function found

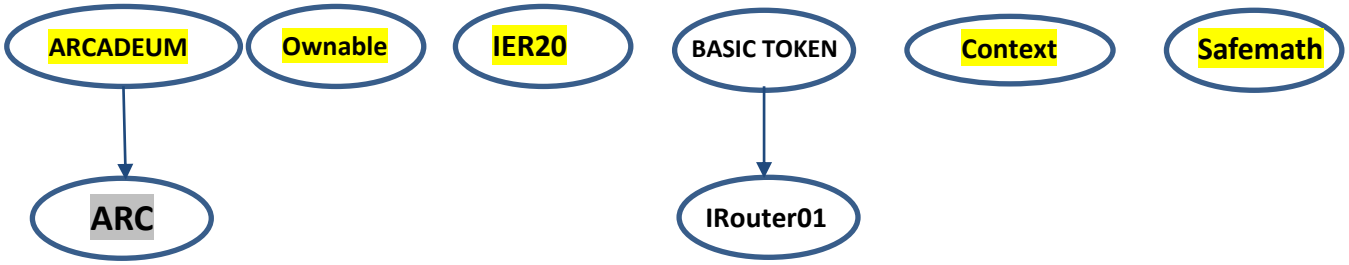
Finding Summary



Status Icon Definitions

✅	Resolved	🚧	In Progress	ℹ️	Ignored (pro)
❌	Not Resolved	❓	Incorrect	🚫	Ignored (con)

INHERITANCE GRAPH



Identifier	Definition	Severity
CEN-12	Centralization privileges of ARCADEUM	Medium #

Vulnerability 0 : No important security issue detected.
Threat level: Low

Home | Arcadeum.sol

```
86
87      address public immutable WETH;
88      IWETH9 public immutable weth;
89      INonfungiblePositionManager public immutable nonfungiblePositionManager;
90      ILPFeeReceiver public lpFeeReceiver;
91
92      uint256 public tokenId;
93
94      uint256 public amount0Collected;
95      uint256 public amount1Collected;
96
97      event FeesCollected(uint256 indexed _amount0, uint256 indexed _amount1, address _lpFeeReceiver, uint256 indexed _timestamp);
98
99      constructor (address _WETH, address _nonfungiblePositionManager, address _lpFeeReceiver) ERC20("Arcadeum", "ARC") {
100         WETH = _WETH;
101         weth = IWETH9(_WETH);
102         nonfungiblePositionManager = INonfungiblePositionManager(_nonfungiblePositionManager);
103         lpFeeReceiver = ILPFeeReceiver(_lpFeeReceiver);
104         _mint(_msgSender(), 10000000 * (10**18));
105     }
106
107     function claim() external nonReentrant {
108         if (tokenId == 0) {
109             revert TokenIdNotSet();
110         }
111
```

Vulnerability Scan

REENTRANCY

Severity	Major
Confidence Parameter	Certain

Vulnerability Description

NOTE: In a re-entrance attack, a malicious contract calls back into the calling contract before the first invocation of the function is finished. This may cause the different invocations of the function to interact in undesirable ways, especially in cases where the function is updating state variables after the external calls.
This may lead to loss of funds, improper value updates, token loss, etc.

Scanning Line:

```
contract ARC is ERC20, IERC721Receiver, Ownable,
ReentrancyGuard {
    error TokenIdNotSet();

    address public immutable WETH;
    IWETH9 public immutable weth;
    INonfungiblePositionManager public immutable
nonfungiblePositionManager;
    ILPFeeReceiver public lpFeeReceiver;

    uint256 public tokenId;

    uint256 public amount0Collected;
    uint256 public amount1Collected;

    event FeesCollected(uint256 indexed _amount0, uint256
indexed _amount1, address _lpFeeReceiver, uint256 indexed
_timestamp);

    constructor (address _WETH, address
_nonfungiblePositionManager, address _lpFeeReceiver)
ERC20("Arcadeum", "ARC") {
        WETH = _WETH;
        weth = IWETH9(_WETH);
        nonfungiblePositionManager =
INonfungiblePositionManager(_nonfungiblePositionManager);
        lpFeeReceiver = ILPFeeReceiver(_lpFeeReceiver);
        _mint(_msgSender(), 10000000 * (10**18));
```

Scanning Line:

```
 function claim() external nonReentrant {
        if (tokenId == 0) {
            revert TokenIdNotSet();
        }

        (uint256 _amount0, uint256 _amount1) =
nonfungiblePositionManager.collect(INonfungiblePositionManager.Colle
ctParams({tokenId
            : tokenId,
            recipient: address(this),
            amount0Max: type(uint128).max,
            amount1Max: type(uint128).max
        }));

        _burn(address(this), _amount0);
        weth.withdraw(_amount1);

        lpFeeReceiver.depositYield{value: address(this).balance}();

        amount0Collected += _amount0;
        amount1Collected += _amount1;
        emit FeesCollected(_amount0, _amount1,
address(lpFeeReceiver), block.timestamp);
```

Recommen-dation:

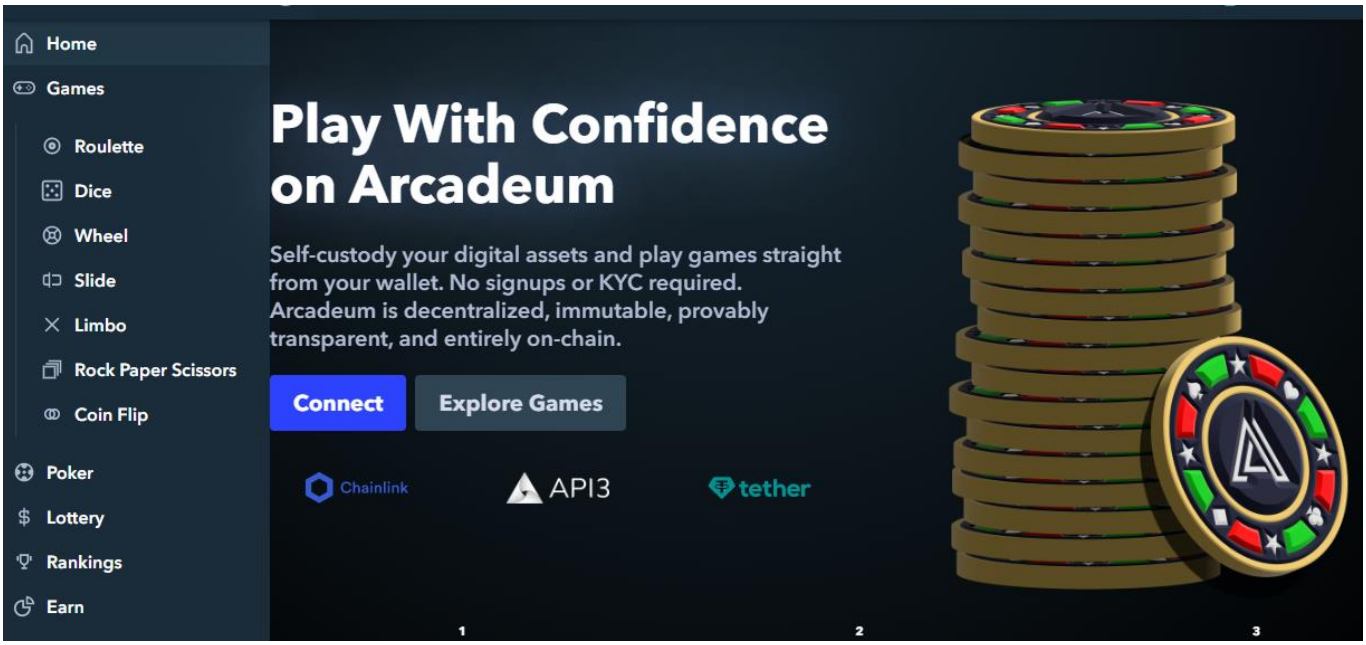
It is recommended to add a **[https://docs.openzeppelin.com/con-tracts/4.x/api/security#ReentrancyGuard]** to the functions making external calls. The functions should use a Checks-Effects-Interactions pattern. The external calls should be executed at the end of the function and all the state-changing must happen before the call..

MANUAL REVIEW

ARCADEUM: is the GMX of on-chain betting. Provable fair quantum RNG. Up to 999x payouts. Stake ARC for fees. Self-custody your digital assets and play games straight from your wallet. No signups or KYC required. Arcadeum is decentralized, immutable, provably transparent, and entirely on-chain.

ARBISHIELD: ARCADEUM
Ticker: ARC
Decimals: 18
Chain/Standard: Arbitrum Network

Outstanding Features oF ARCADEUM Launching On Arbitrum Network



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

```
    function onERC721Received(address, address, uint256, bytes calldata) external override
returns (bytes4) {
        return this.onERC721Received.selector;
    }

    receive() external payable {}
    }
```

Description:

Floating point calculations can vary across different architectures.

Recommendation: Replace with sdk.Dec .

Alleviation:

This exhibit was acknowledged and ultimately discarded by the Arcadeum team due to low severity. We consider the exhibit fully attended to as it doesn't impose any meaningful security concerns.

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

ARCADEUM project team understands the centralization risk. Some functions are provided privileged access to ensure a good runtime behaviour in the project

Identifier	Definition	Severity
COD-10	Third Party Dependencies	Minor

A smart contract is interacting with third-party protocols e.g., Uniswap, Pancakeswap router, cashier contract,

And 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.

DISCLAIMERS

Vital Block Security 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 OTHER WEBSITES

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://www.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

Vital Block Dedicated to securing Public and Private Blockchain Ecosystem