CERTIK

Code Security Assessment

Axes Metaverse

Feb 24th, 2022

Table of Contents

Summary

This report has been prepared for Axes Metaverse to discover issues and vulnerabilities in the source code of the Axes Metaverse project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Axes Metaverse
Description	DEX
Platform	BSC
Language	Solidity
Codebase	https://ropsten.etherscan.io/address/0xECE4cD3b324C40144CA153F67B998C56E3De8BE2#code https://ropsten.etherscan.io/address/0x87FA9690b5817Dd90492272e659D2F3C3E079Ae7#code
Commit	N/A

Audit Summary

Delivery Date	Feb 24, 2022
Audit Methodology	Static Analysis, Manual Review

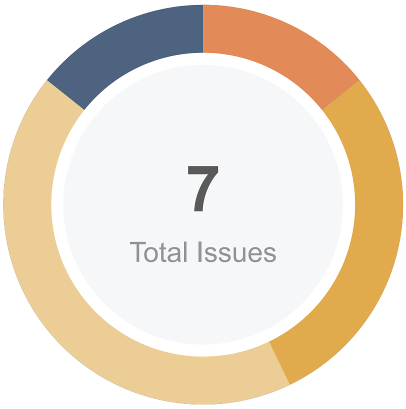
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
● Critical	0	0	0	0	0	0	0
● Major	1	0	0	1	0	0	0
● Medium	2	0	0	0	0	0	2
● Minor	3	0	0	0	0	0	3
● Informational	1	0	0	0	0	0	1
● Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
AMC	AMCmarketplace.sol	be979ea510f577f4df8dbb751d2b19fa9233f55c23060c98f600c0cfa1ef638b
M0X	m.sol	a1b03113dc3491d3acbbd8c5b629fe4df5d608a3f8ff86645bfca2cf6add8ef8

Findings



🟥	**Critical**	**0** (0.00%)
🟧	**Major**	**1** (14.29%)
🟨	**Medium**	**2** (28.57%)
🟨	**Minor**	**3** (42.86%)
🟦	**Informational**	**1** (14.29%)
🟩	**Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
AMC-01	Fee Can Be Changed After Posting A Sale Request	Control Flow	● Medium	✓ Resolved
AMC-02	Ignore Return Values	Volatile Code	● Medium	✓ Resolved
AMC-03	Third Party Dependencies	Volatile Code	● Minor	✓ Resolved
AMC-04	No Upper Limit for `fee`	Control Flow	● Minor	✓ Resolved
AMC-05	Missing Input Validation	Control Flow	● Informational	✓ Resolved
AMC-06	Arbitrary External Call	Volatile Code	● Minor	✓ Resolved
M0X-01	Centralization Related Risks	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged

AMC-01 | Fee Can Be Changed After Posting A Sale Request

Category	Severity	Location	Status
Control Flow	● Medium	AMCmarketplace.sol (1): 1650	✓ Resolved

Description

In function `buy()`:

```
1650        uint256 _fee = sale[_id].price.div(denominator).mul(fee);
```

When users put an ERC721 token on sale, the amount of ERC20 that they expect to receive is the sale price minus the current fee. However, if the fee is raised before someone buys the token, the seller will receive fewer ERC20 tokens than the amount when he listed it.

Recommendation

We recommend the team be transparent regarding the behavior. One potential solution is adding the current `denominator` and `fee` to the `Sales` struct when a user calls `sell()`.

Alleviation

The team heeded our advice and resolved the issue in the new version.

AMC-02 | Ignore Return Values

Category	Severity	Location	Status
Volatile Code	● Medium	AMCmarketplace.sol (1): 1651~1652	✓ Resolved

Description

In function `buy()`:

```
1651        IERC20(sale[_id].c20).transferFrom(_msgSender(), sale[_id].seller, sale[_id].price.sub(_fee));
1652        IERC20(sale[_id].c20).transferFrom(_msgSender(), feeAddr, _fee);
```

Function `buy()` ignores return values by the function calls of `transferFrom()`.

Several tokens do not revert in case of failure and return false. The buyer could get the ERC721 token without paying ERC20 tokens.

Recommendation

We recommend handling the return values.

Alleviation

The team heeded our advice and resolved the issue in the new version.

AMC-03 | Third Party Dependencies

Category	Severity	Location	Status
Volatile Code	● Minor	AMCmarketplace.sol (1): 1563	✓ Resolved

Description

```
1555     interface IQbeinlicensing {
1556        function checkLicense(address _contract) external view returns (bool active);
1557     }
1558
1559     contract Qbeinlicensing {
1560         address public licensingContract;
1561
1562         constructor(address _license) {
1563             licensingContract = _license;
1564         }
1565
1566         modifier isRunning {
1567             require(IQbeinlicensing(licensingContract).checkLicense(address(this)),
'License not active.');
1568             _;
1569         }
1570     }
```

The contract is serving as the underlying entity to interact with third party contract `licensingContract`. The scope of the audit treats 3rd party entities as black boxes and assumes their functional correctness. However, in the real world, 3rd parties can be compromised and this may lead to transaction failure since major functions in contract `AMCmarketplace` are guarded by the modifier `isRunning`.

Recommendation

We understand that the business logic of `AMCmarketplace` requires interaction with the contract that implements interface `IQbeinlicensing`. We encourage the team to constantly monitor the statuses of the 3rd party to mitigate the side effects when unexpected activities are observed.

Alleviation

The team removed the code related to the third party in the new version.

AMC-04 | No Upper Limit For `fee`

Category	Severity	Location	Status
Control Flow	● Minor	AMCmarketplace.sol (1): 1674~1681	✓ Resolved

Description

```
    function changeFee(uint256 _fee, uint256 _denominator, address _feeAddr) external isRunning {
        require(hasRole(DEFAULT_ADMIN_ROLE, _msgSender()), "You must have admin role to change fee.");
        require(_denominator >= _fee, "Fee is more than denominator.");
        fee = _fee;
        denominator = _denominator;
        feeAddr = _feeAddr;
        emit ChangeFee(_fee, _denominator, _feeAddr);
    }
```

There is no upper limit restricting parameter `_fee` of `changeFee()` potentially enabling up to 100% fees on transactions.

Recommendation

We recommend setting an upper limit for the fee variable.

Alleviation

The team heeded our advice and set a maximum 30% fee in the new version.

AMC-05 | Missing Input Validation

Category	Severity	Location	Status
Control Flow	● Informational	AMCmarketplace.sol (1): 1679	✓ Resolved

Description

Function `ChangeFee()` does not check if address `_feeAddr` is 0.

Recommendation

We recommend adding zero-address check for address `_feeAddr`.

Alleviation

The team heeded our advice and resolved the issue in the new version.

AMC-06 | Arbitrary External Call

Category	Severity	Location	Status
Volatile Code	● Minor	AMCmarketplace.sol (1): 1625	✓ Resolved

Description

```
  1624      function sell(address _contract721, address _contract20, uint256 _tokenId, address
_from, uint256 _price) external isRunning returns (uint256 _id) {
  1625          IERC721(_contract721).safeTransferFrom(_from, address(this), _tokenId);
  1626          require(_price != 0, "Price cannot be zero.");
  1627          require(_price % denominator == 0, "Price in wei must be a multiple of
denominator. Remember the decimals.");
  1628          require(contractsWhitelist.contains(_contract721), "ERC721 contract is not in
whitelist.");
  1629          require(contractsWhitelist.contains(_contract20), "ERC20 contract is not in
whitelist.");
  1630          sellId = sellId.add(1);
  1631          onSale = onSale.add(1);
  1632          idByToken[_contract721][_tokenId] = sellId;
  1633          sale[sellId] = Sales(_msgSender(), address(0), _contract721, _contract20,
_tokenId, _price, true);
  1634          emit Sell(sellId, _tokenId, _price, _from, _contract721, _contract20);
  1635          return sellId;
  1636      }
```

The function `sell()` calls external function `IERC721(_contract721).safeTransferFrom()` before checking if `contractsWhitelist` contains `_contract721`.

Recommendation

We recommend checking if `contractsWhitelist` contains the ERC721 contract before calling its function.

Alleviation

The team heeded our advice and resolved the issue in the new version.

M0X-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	m.sol (2): 1634~1654	ⓘ Acknowledged

Description

```
  1634      function addToWhitelist(address _contract) external {
  1635          require(hasRole(DEFAULT_ADMIN_ROLE, _msgSender()), "You must have admin role to
add contract to whitelist.");
  1636          require(_contract.isContract(), "Address is not a contract.");
  1637          require(contractsWhitelist.add(_contract), "Contact is already in whitelist.");
  1638          emit ChangeContractWhiteList(_contract, true);
  1639      }
  1640
  1641      function removeFromWhitelist(address _contract) external {
  1642          require(hasRole(DEFAULT_ADMIN_ROLE, _msgSender()), "You must have admin role to
remove contract from whitelist.");
  1643          require(contractsWhitelist.remove(_contract), "Where is no such contract in
whitelist.");
  1644          emit ChangeContractWhiteList(_contract, false);
  1645      }
  1646
  1647      function changeFee(uint256 _fee, address _feeAddr) external {
  1648          require(hasRole(DEFAULT_ADMIN_ROLE, _msgSender()), "You must have admin role to
change fee.");
  1649          require(_fee <= 300, "Fee cannot be more than 30%.");  // 30%, denominator is
constant 1000
  1650          require(_feeAddr != address(0), "Wrong fee address.");
  1651          fee = _fee;
  1652          feeAddr = _feeAddr;
  1653          emit ChangeFee(_fee, _feeAddr);
  1654      }
```

In the contract `AMCmarketplace` the role `DEFAULT_ADMIN_ROLE` has authority over the functions above.

Any compromise to the `DEFAULT_ADMIN_ROLE` account may allow a hacker to take advantage of this authority, remove contracts from the white list, change fee rates, and/or change the fee receiver.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement;
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles;
  OR
- Remove the risky functionality.

Noted: Recommend considering the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

Alleviation

[Axes Metaverse]: We plan to use multisig wallet as admin to mitigate this risk

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you (“Customer” or the “Company”) in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. 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 CertiK’s prior written consent in each instance.

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 considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts CertiK to perform a security assessment. This report 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 in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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. CertiK’s position is that each company and individual are responsible for their own due diligence and continuous security. CertiK’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 CertiK is subject to dependencies and 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.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR 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, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'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. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

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

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.