

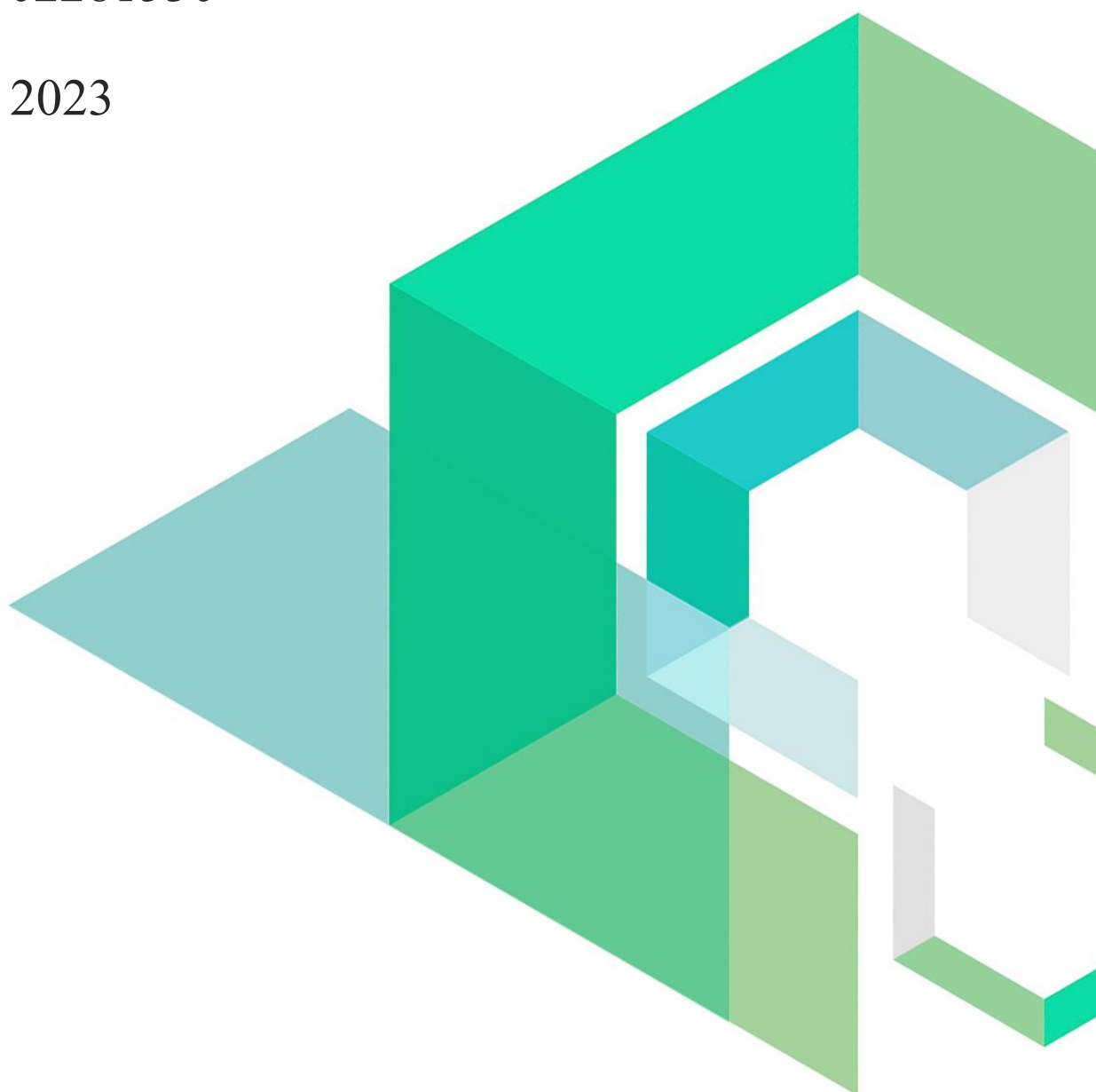
AMC

Smart Contract Security Audit

V1.0

No. 202302281530

Feb 28th, 2023

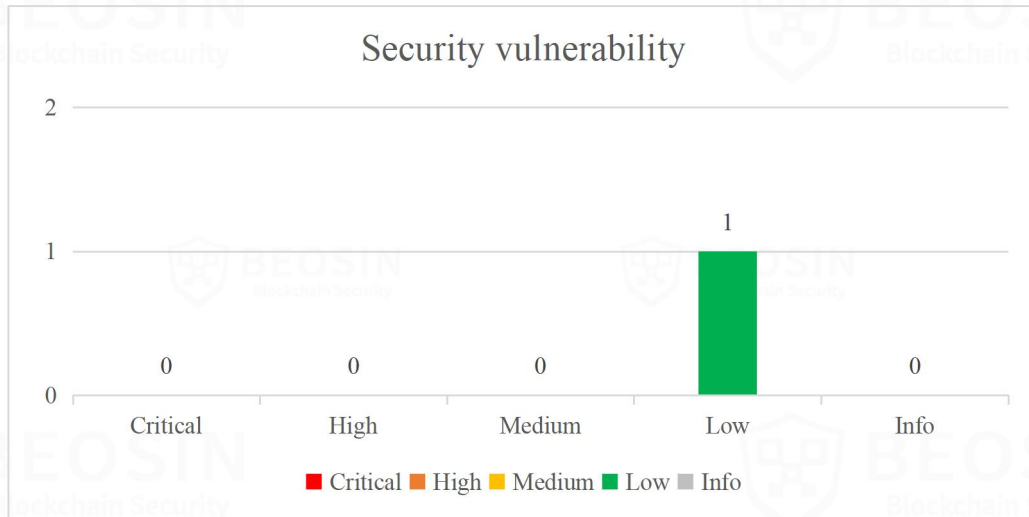


Contents

| | |
|--|----------|
| Summary of Audit Results | 1 |
| 1 Overview | 3 |
| 1.1 Project Overview | 3 |
| 1.2 Audit Overview | 3 |
| 2 Findings | 4 |
| [AMC-1] Centralization risk | 5 |
| 3 Appendix | 7 |
| 3.1 Vulnerability Assessment Metrics and Status in Smart Contracts | 7 |
| 3.2 Audit Categories | 10 |
| 3.3 Disclaimer | 12 |
| 3.4 About Beosin | 13 |

Summary of Audit Results

After auditing, 1 Low risk items were identified in the AMC project. Specific audit details will be presented in the **Findings** section. Users should pay attention to the following aspects when interacting with this project:



*Notes:

● Risk Description:

1. There is asset centralization risk that this project mint one billion token to deployer's address.

Project Description:

1. Basic Token Information

| | |
|--------------|--------------------------------|
| Token name | AI Meta Coin |
| Token symbol | AMC |
| Decimals | 18 |
| Pre-mint | 1,000,000,000(All to deployer) |
| Total supply | 1,000,000,000 |
| Token type | ERC20 |

Table 1 Basic information of AMC

2. Business overview

The AMC project is based on ERC20 token that deployed on Arbitrum One chain. The total supply of token is one billion which is minted to deployer's address when the contract is deployed. After contract deployed, AMC token can't be minted and can't burned.

1 Overview

1.1 Project Overview

| | |
|-------------------------|--|
| Project Name | AMC |
| Platform | Arbitrum One |
| Contract Address | 0x299142a6370e1912156e53fbd4f25d7ba49ddcc5 |

1.2 Audit Overview

Audit work duration: Feb 27, 2023 –Feb 28, 2023

Audit methods: Formal Verification, Static Analysis, Typical Case Testing and Manual Review.

Audit team: Beosin Security Team.

2 Findings

| Index | Risk description | Severity level | Status |
|-------|---------------------|----------------|--------|
| AMC-1 | Centralization risk | Low | Fixed |

Finding Details:

[AMC-1] Centralization risk

| | |
|----------------|--|
| Severity Level | Low |
| Type | Business Security |
| Lines | AlMetaCoin.sol.sol #L546-549 |
| Description | After contract creation, the total supply of one billion tokens are allocated to the deployer's account through the <code>_mint</code> function, which has the risk of centralization of token allocation. |

```

541
542     pragma solidity ^ 0.8 .9;
543
544
545     contract AlMetaCoin is ERC20 {
546         constructor() ERC20("AI Meta Coin", "AMC") {
547             _mint(msg.sender, 1000000000 * 10 ** decimals());
548         }
549     }

```

Figure 1 Source code of *constructor* function

| | |
|-----------------|---|
| Recommendations | It is recommended to use multi-signature wallet or DAO governance to manage admin and minter. |
|-----------------|---|

Fixed. The project team transfer all token to GnosisSafeProxy address (0x71024D7938Fc99F108Cb16674A1E4E0C7d6b865e) which is multi-signature wallet. This GnosisSafeProxy address setup 2 owner address to manage this wallet.

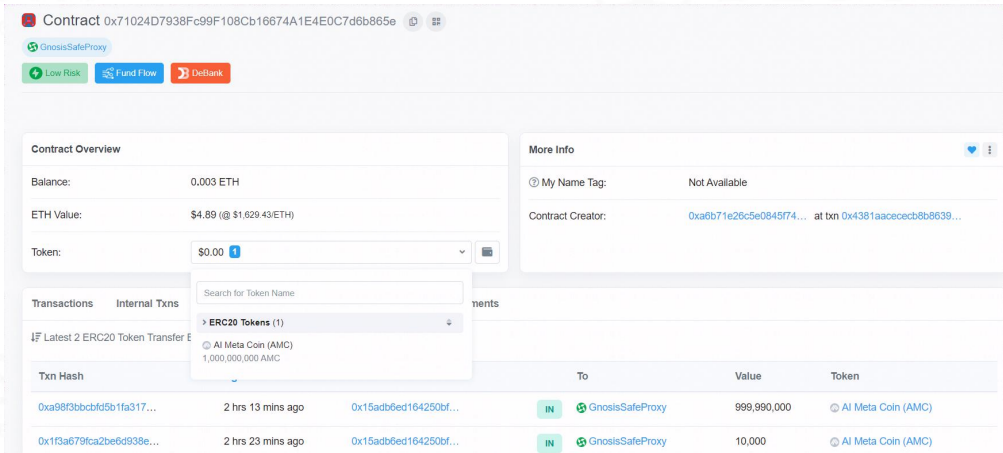
| | |
|--------|---|
| Status |  <p>The screenshot shows the GnosisSafeProxy contract overview for address 0x71024D7938Fc99F108Cb16674A1E4E0C7d6b865e. It displays a balance of 0.003 ETH and an ETH value of \$4.89. The token is AI Meta Coin (AMC) with a value of \$0.00. The 'Transactions' section shows two recent transactions where 999,990,000 and 10,000 tokens were transferred to the GnosisSafeProxy address from the contract creator.</p> |
|--------|---|

Figure 2 GnosisSafeProxy multi-signature wallet address transaction



Figure 3 Two owner address for GnosisSafeProxy multi-signature wallet

3 Appendix

3.1 Vulnerability Assessment Metrics and Status in Smart Contracts

3.1.1 Metrics

In order to objectively assess the severity level of vulnerabilities in blockchain systems, this report provides detailed assessment metrics for security vulnerabilities in smart contracts with reference to CVSS 3.1 (Common Vulnerability Scoring System Ver 3.1).

According to the severity level of vulnerability, the vulnerabilities are classified into four levels: "critical", "high", "medium" and "low". It mainly relies on the degree of impact and likelihood of exploitation of the vulnerability, supplemented by other comprehensive factors to determine of the severity level.

| Impact Likelihood | Severe | High | Medium | Low |
|----------------------|----------|--------|--------|------|
| Probable | Critical | High | Medium | Low |
| Possible | High | High | Medium | Low |
| Unlikely | Medium | Medium | Low | Info |
| Rare | Low | Low | Info | Info |

3.1.2 Degree of impact

- **Severe**

Severe impact generally refers to the vulnerability can have a serious impact on the confidentiality, integrity, availability of smart contracts or their economic model, which can cause substantial economic losses to the contract business system, large-scale data disruption, loss of authority management, failure of key functions, loss of credibility, or indirectly affect the operation of other smart contracts associated with it and cause substantial losses, as well as other severe and mostly irreversible harm.

- **High**

High impact generally refers to the vulnerability can have a relatively serious impact on the confidentiality, integrity, availability of the smart contract or its economic model, which can cause a

greater economic loss, local functional unavailability, loss of credibility and other impact to the contract business system.

- **Medium**

Medium impact generally refers to the vulnerability can have a relatively minor impact on the confidentiality, integrity, availability of the smart contract or its economic model, which can cause a small amount of economic loss to the contract business system, individual business unavailability and other impact.

- **Low**

Low impact generally refers to the vulnerability can have a minor impact on the smart contract, which can pose certain security threat to the contract business system and needs to be improved.

3.1.4 Likelihood of Exploitation

- **Probable**

Probable likelihood generally means that the cost required to exploit the vulnerability is low, with no special exploitation threshold, and the vulnerability can be triggered consistently.

- **Possible**

Possible likelihood generally means that exploiting such vulnerability requires a certain cost, or there are certain conditions for exploitation, and the vulnerability is not easily and consistently triggered.

- **Unlikely**

Unlikely likelihood generally means that the vulnerability requires a high cost, or the exploitation conditions are very demanding and the vulnerability is highly difficult to trigger.

- **Rare**

Rare likelihood generally means that the vulnerability requires an extremely high cost or the conditions for exploitation are extremely difficult to achieve.

3.1.5 Fix Results Status

| Status | Description |
|-----------------|--|
| Fixed | The project party fully fixes a vulnerability. |
| Partially Fixed | The project party did not fully fix the issue, but only mitigated the issue. |

Acknowledged

The project party confirms and chooses to ignore the issue.

3.2 Audit Categories

| No. | Categories | Subitems |
|-----|-----------------------|--|
| 1 | Coding Conventions | Compiler Version Security |
| | | Deprecated Items |
| | | Redundant Code |
| | | require/assert Usage |
| | | Gas Consumption |
| 2 | General Vulnerability | Integer Overflow/Underflow |
| | | Reentrancy |
| | | Pseudo-random Number Generator (PRNG) |
| | | Transaction-Ordering Dependence |
| | | DoS (Denial of Service) |
| | | Function Call Permissions |
| | | call/delegatecall Security |
| | | Returned Value Security |
| | | tx.origin Usage |
| | | Replay Attack |
| | | Overriding Variables |
| | | Third-party Protocol Interface Consistency |
| 3 | Business Security | Business Logics |
| | | Business Implementations |
| | | Manipulable Token Price |
| | | Centralized Asset Control |
| | | Asset Tradability |
| | | Arbitrage Attack |

Beosin classified the security issues of smart contracts into three categories: Coding Conventions, General Vulnerability, Business Security. Their specific definitions are as follows:

- **Coding Conventions**

Audit whether smart contracts follow recommended language security coding practices. For example, smart contracts developed in Solidity language should fix the compiler version and do not use deprecated keywords.

- **General Vulnerability**

General Vulnerability include some common vulnerabilities that may appear in smart contract projects. These vulnerabilities are mainly related to the characteristics of the smart contract itself, such as integer overflow/underflow and denial of service attacks.

- **Business Security**

Business security is mainly related to some issues related to the business realized by each project, and has a relatively strong pertinence. For example, whether the lock-up plan in the code match the white paper, or the flash loan attack caused by the incorrect setting of the price acquisition oracle.

*Note that the project may suffer stake losses due to the integrated third-party protocol. This is not something Beosin can control. Business security requires the participation of the project party. The project party and users need to stay vigilant at all times.

3.3 Disclaimer

The Audit Report issued by Beosin is related to the services agreed in the relevant service agreement. The Project Party or the Served Party (hereinafter referred to as the "Served Party") can only be used within the conditions and scope agreed in the service agreement. Other third parties shall not transmit, disclose, quote, rely on or tamper with the Audit Report issued for any purpose.

The Audit Report issued by Beosin is made solely for the code, and any description, expression or wording contained therein shall not be interpreted as affirmation or confirmation of the project, nor shall any warranty or guarantee be given as to the absolute flawlessness of the code analyzed, the code team, the business model or legal compliance.

The Audit Report issued by Beosin is only based on the code provided by the Served Party and the technology currently available to Beosin. However, due to the technical limitations of any organization, and in the event that the code provided by the Served Party is missing information, tampered with, deleted, hidden or subsequently altered, the audit report may still fail to fully enumerate all the risks.

The Audit Report issued by Beosin in no way provides investment advice on any project, nor should it be utilized as investment suggestions of any type. This report represents an extensive evaluation process designed to help our customers improve code quality while mitigating the high risks in blockchain.

3.4 About Beosin

Beosin is the first institution in the world specializing in the construction of blockchain security ecosystem. The core team members are all professors, postdocs, PhDs, and Internet elites from world-renowned academic institutions. Beosin has more than 20 years of research in formal verification technology, trusted computing, mobile security and kernel security, with overseas experience in studying and collaborating in project research at well-known universities. Through the security audit and defense deployment of more than 2,000 smart contracts, over 50 public blockchains and wallets, and nearly 100 exchanges worldwide, Beosin has accumulated rich experience in security attack and defense of the blockchain field, and has developed several security products specifically for blockchain.



Official Website

<https://www.beosin.com>

Telegram

<https://t.me/+dD8Bnqd133RmNWNl>

Twitter

https://twitter.com/Beosin_com

Email

Contact@beosin.com

