BEOSIN
Blockchain Security

AURX

Smart Contract Security Audit

No. 202606101715

June 10th, 2026

SECURING BLOCKCHAIN ECOSYSTEM

WWW.BEOSIN.COM

Contents

1 Overview 5

1.1 Project Overview 5

1.2 Audit Overview 5

1.3 Audit Method 5

2 Findings 7

[AURX-01] Centralization risk 8

[AURX-02] decimals variable does not comply with the ERC-20 standard 10

3 Appendix 11

3.1 Vulnerability Assessment Metrics and Status in Smart Contracts 11

3.2 Audit Categories 14

3.3 Disclaimer 16

3.4 About Beosin 17

Summary of Audit Results

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

Medium	Fixed:0 Acknowledged:1
Low	Fixed:0 Acknowledged:1

Project Description:

1. Basic Token Information

Token name	AURIX Network
Token symbol	AURX
Token decimals	18
Total Supply	10,000,000,000
Token type	ERC-20

Table 1 ARUX token info

2. Business overview

AURX is an ERC-20 standard token deployed on the Binance Smart blockchain with a fixed total supply of 10 billion tokens. The token supply is non-mintable, meaning no additional tokens can be minted after deployment, but holders are allowed to burn their tokens to reduce the total supply. The contract implements the core ERC-20 interfaces, supporting fundamental functionalities such as transfers, approvals, balance queries, and transfer restrictions. Additionally, the contract includes locking mechanisms and an allowlist system to provide flexible control over token transfers.

1 Overview

1.1 Project Overview

Project Name	AURX
Project Language	Solidity
Platform	Binance Smart Chain
Contract Address	0x24ECb00840081D56116fC6D076988411a5595fd0

1.2 Audit Overview

Audit work duration: June 10, 2026

Audit team: Beosin Security Team

1.3 Audit Method

The audit methods are as follows:

1. Formal Verification

Formal verification is a technique that uses property-based approaches for testing and verification. Property specifications define a set of rules using Beosin's library of security expert rules. These rules call into the contracts under analysis and make various assertions about their behavior. The rules of the specification play a crucial role in the analysis. If the rule is violated, a concrete test case is provided to demonstrate the violation.

2. Manual Review

Using manual auditing methods, the code is read line by line to identify potential security issues. This ensures that the contract's execution logic aligns with the client's specifications and intentions, thereby safeguarding the accuracy of the contract's business logic.

The manual audit is divided into three groups to cover the entire auditing process:

The Basic Testing Group is primarily responsible for interpreting the project's code and conducting comprehensive functional testing.

The Simulated Attack Group is responsible for analyzing the audited project based on the collected historical audit vulnerability database and security incident attack models. They identify potential attack vectors and collaborate with the Basic Testing Group to conduct simulated attack tests.

The Expert Analysis Group is responsible for analyzing the overall project design, interactions with third parties, and security risks in the on-chain operational environment. They also conduct a review of the entire audit findings.

3. Static Analysis

Static analysis is a method of examining code during compilation or static analysis to detect issues. Beosin-VaaS can detect more than 100 common smart contract vulnerabilities through static analysis, such as reentrancy and block parameter dependency. It allows early and efficient discovery of problems to improve code quality and security.

2 Findings

Index	Risk description	Severity level	Status
AURX-01	Centralization risk	Medium	Acknowledged
AURX-02	decimals variable does not comply with the ERC-20 standard	Low	Acknowledged

Finding Details:

[AURX-01] Centralization risk

Severity Level	Medium
Type	Business Security
Lines	/contracts/IRCStaking.sol #L218
Description	We have identified two major centralization risks in the AURX token contract: (1) The contract inherits from Ownable, and the contract owner (owner) has the authority to enable the locked state. When enabled, only whitelisted addresses are allowed to transfer tokens. This means the owner can arbitrarily restrict any user from transferring their tokens. Furthermore, the current owner address (0x55478519Fb44D3f52FE4DdEbA5d1D57856B4cc37) is an Externally Owned Account (EOA), which introduces a centralization risk — if the private key of this address is lost or compromised, it may prevent users from transferring tokens normally. All three are EOAs. This concentration of token holdings poses a substantial centralization risk. If any of these addresses are compromised or the private keys are lost, it could negatively impact the token's circulation, market stability, and user trust.
Recommendation	Introduce Multisig ownership and token distribution Strategy (1) Replace the EOA owner address with a multisignature wallet It is recommended to replace the current EOA-based contract owner with a multisignature wallet (e.g., Gnosis Safe). This change would require multiple signatures to perform sensitive operations, significantly reducing the risk of a single point of failure and improving governance security and transparency. (2) Redesign the token distribution to avoid concentration A large proportion of tokens are currently held by a few EOA addresses. It is recommended to adopt a more decentralized distribution strategy before the token goes live. This may include team vesting contracts, community incentive

	pools, and foundation-managed reserves to gradually.
Status	**Acknowledged.** Therefore, although this issue has not been directly addressed, the project team clarified that the locking mechanism in the contract is intended solely for operational control of institutional token holdings and is not meant to restrict regular users.

[AURX-02] decimals variable does not comply with the ERC-20 standard

Severity Level	Low
Type	Coding Conventions
Lines	Token.sol #L284
Description	According to the ERC-20 standard, the decimals() function must return a value of type uint8. However, this contract uses uint (i.e., uint256) instead. This mismatch may lead to decoding failures or runtime exceptions when interacting with contracts or DApps that rely on strict ABI compliance. `uint public constant decimals = 18;`
Recommendation	It is recommended to Update the type of the decimals variable from uint to uint8 to comply with the ERC-20 standard.
Status	**Acknowledged.**

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.

Likelihood \ Impact	Severe	High	Medium	Low
Probable	Critical	High	Medium	Low
Possible	High	Medium	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.

3.1.3 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.

3.1.4 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.

BEOSIN
Blockchain Security

Official Website
https://www.beosin.com

Telegram
https://t.me/beosin

Twitter
https://twitter.com/Beosin_com

Email
service@beosin.com