BEOSIN
Web3 Security & Compliance

Nudix

Smart Contract Security Audit

No. 202505291029

May 29th, 2025

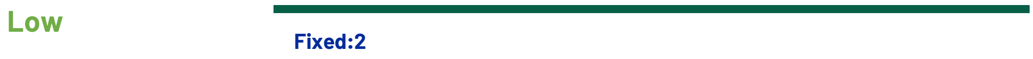
SECURING BLOCKCHAIN ECOSYSTEM

WWW.BEOSIN.COM

Contents

Summary of Audit Result

After auditing, 2 Low risk items were identified in the Nudix project. Specific audit details will be presented in the Findings section. Users should pay attention to the following aspects when interacting with this project:



Business overview

Token name	Temporary Nudix
Token symbol	T-NUDIX
Decimals	18
Initial supply	0
Token type	BEP-20

Table 1 T-NUDIX token info

T-NUDIX is planned to be deployed as a BEP-20 token on the BNB Smart Chain. It has an initial supply of zero and supports both minting and burning of tokens, with a maximum minting limit of one billion tokens. Only users with the MINTER role are allowed to mint new tokens, while burning tokens is restricted to users who are whitelisted. In addition to inheriting the standard BEP-20 functionalities, T-NUDIX also modifies the token transfer mechanism by requiring that the recipient's address be whitelisted before any transfer can occur.

The NudixSale contract is specifically designed for selling T-NUDIX tokens. The contract administrator can start each sale round by setting the sale price, minimum purchase amount, and maximum sale cap. Users can then use USDT to purchase T-NUDIX tokens. If the total amount of tokens sold in the current round reaches the sale cap, the contract will automatically finalize the round, or the administrator can manually end it. A new sale round can only be initiated once the current one has ended.

1 Overview

1.1 Project Overview

Project Name	Nudix
Project Language	Solidity
Platform	BNB Smart Chain
Code Base	https://github.com/Nudix2/contracts
Commit ID	d4c133931a1cf218213cd7ca749518170fe35424 2f0426dff53c45c0dd7bc7d07766382deee9a80f

1.2 Audit Overview

Audit work duration: May 26, 2025 - May 27, 2025, May 29, 2025

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
Nudix-01	Centralization Risk	Low	Fixed
Nudix-02	Purchase Logic Can Be Optimized	Low	Fixed

Finding Details:

[Nudix-01] Centralization Risk

Severity Level	Low
Type	Business Security
Code Location	/src/TemporaryNudix.sol #L49, L58
Description	Based on the current code logic, the `T-NUDIX` token requires the `MINTER_ROLE` to be granted to the Sale contract, allowing tokens to be minted when users purchase them via the Sale contract. This is a good design choice that ensures controlled minting of tokens within the contract. However, there are some associated security risks: 1. No minting cap for `T-NUDIX` tokens: The token contract does not set an upper limit for minting, which means the holder of the MINTER_ROLE can mint an unlimited number of `T-NUDIX` tokens. The risks of unlimited minting include: ① If minting is not capped, additional issuance could dilute existing holders' share of the total supply. ② Exceeding the intended supply might interfere with the economic model and put pressure on the token price. ③ An attacker with MINTER privileges could mint excess tokens, undermining supply discipline and damaging market confidence. 2. DEFAULT_ADMIN controls the MINTER_ROLE, creating a single point of failure: The DEFAULT_ADMIN account has the ability to manage the MINTER_ROLE, granting or revoking it to other addresses at any time, which could compromise the intended token-supply controls.
Recommendation	1. Set a maximum minting cap In the `T-NUDIX` token contract, introduce a global maximum supply variable (such as MAX_SUPPLY) and enforce a check in the _mint function to ensure that the total supply after minting does not exceed this cap. This measure prevents unlimited minting of tokens. 2. Use a multisig wallet for managing `MINTER_ROLE` and `DEFAULT_ADMIN` Delegate the control of `DEFAULT_ADMIN_ROLE` and `MINTER_ROLE` to a

	multisig wallet (such as Gnosis Safe) to reduce the risk of a single point of failure and improve the security of role management.
Status	**Fixed.** The new version of the code has set the minting cap for the `T-NUDIX` token at one billion tokens. Regarding the suggestion to use a multi-signature wallet for project authority management, the project team has stated that they plan to implement this after the project goes live.

[Nudix-02] Purchase Logic Can Be Optimized

Severity Level	Low
Type	Business Security
Code Location	/src//NudixSale.sol #L67-102
Description	In the `NudixSale` contract, purchase amounts must exceed the `minPurchase` threshold and ensure that the total investment after the purchase does not exceed the `roundCap`. This logic can leave a small remainder unsold and may delay the automatic closure of the round.
Recommendation	It is recommended to adjust the user's purchase down to the remaining amount and then closing the round automatically.
Status	**Fixed.** This issue has been fixed in commit d4c13393. Now, when the remaining purchasable amount is less than the `minPurchase` threshold, the current round will automatically end.

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	Medium	Medium	Low
Unlikely	Medium	Medium	Low	Info
Rare	Low	Low	Info	Info

2.1.2 Degree of impact

- **Critical**

Critical 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.

2.1.3 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.

2.1.4 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	SPL Token Standards
		Visibility Specifiers
		Lamport Check
		Account Check
		Signer Check
		Program Id Check
		Deprecated Items
		Redundant Code
2	General Vulnerability	Integer Overflow/Underflow
		Reentrancy
		Pseudo-random Number Generator (PRNG)
		Transaction-Ordering Dependence
		DoS (Denial of Service)
		Function Call Permissions
		Returned Value Security
		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

Web3 Security & Compliance

Official Website

https://www.beosin.com

Telegram

https://t.me/beosin

X

https://x.com/Beosin_com

Email

service@beosin.com

LinkedIn

https://www.linkedin.com/company/beosin/