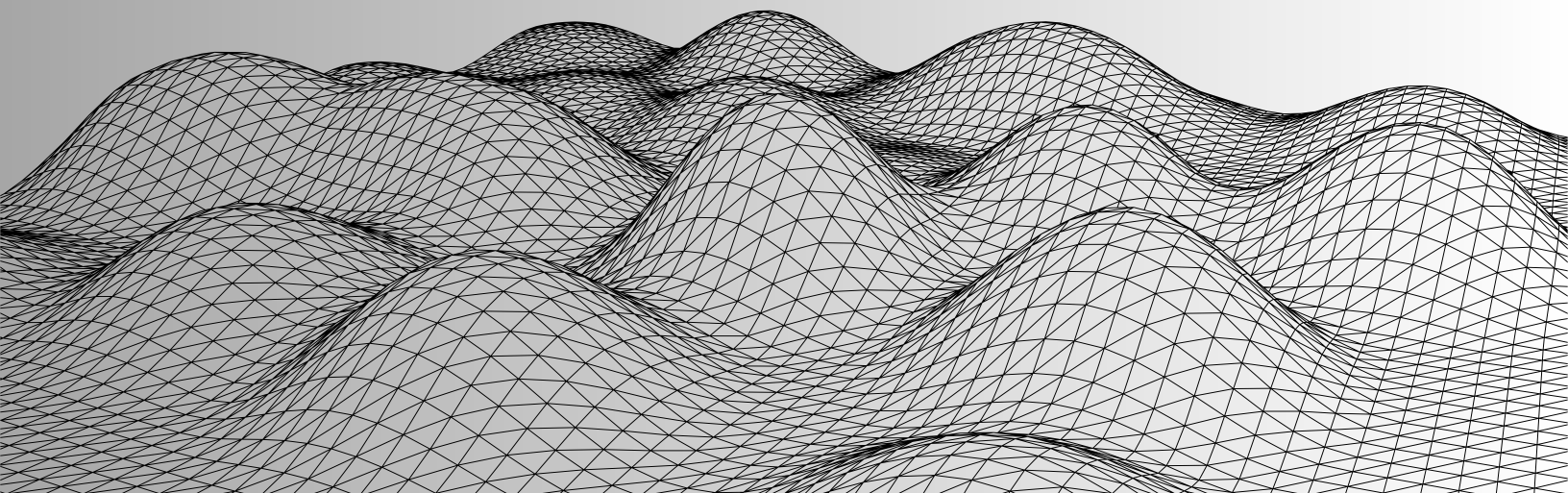
BlockGenesys

Smart Contract Security
Audit

KOI (KOAI)
Token

13 August, 2024



Introduction

This report may include confidential information regarding the IT systems and intellectual property of the Client, along with details on potential vulnerabilities and their possible exploitation. Public disclosure of this report is only permitted with prior consent from the Client. Any further distribution or publication of this report must be authorized by the Client.

Name	Block Genesys
Website	https://blockgenesys.com
Repository/Source	KOI Token
Commit	-
Platform	L1
Network	Ethereum
Languages	Solidity
Timeline	10 Aug 2024 - 13 Aug 2024

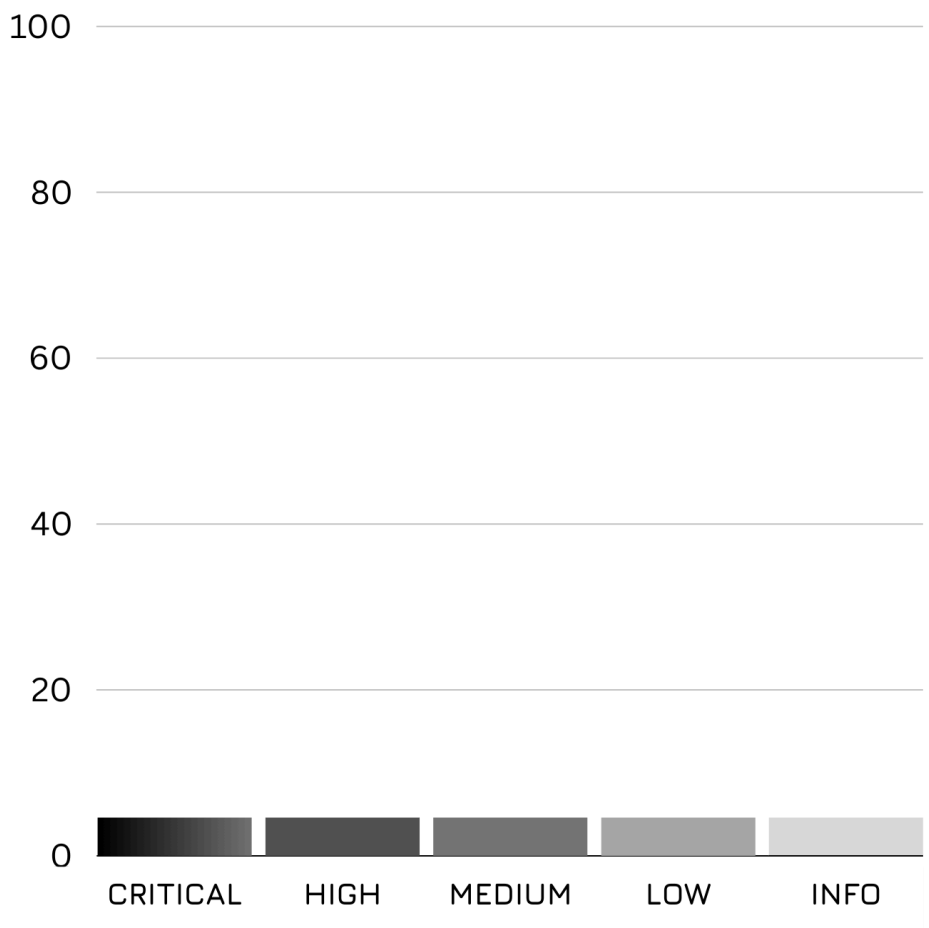
Table of Contents

Introduction 1
Table of Contents 2
Executive Summary 3
Issues Overview 3
Project Overview 4
Core Functionality 4
Token Extensions 4
Utility Libraries 4
Conclusion 4
Scope 5
Out of Scope 5
Methodology 6
Security Review Report 7
Issues found 7
Summary of Findings 8
Findings Overview 9
Disclaimer 10

Executive Summary

Our team performed a technique called “Filtered Security Review”, where the KOI codebase was separately reviewed. After a thorough and rigorous process involving manual code review, automated testing was carried out using; Slither for static analysis and Foundry for fuzzing invariants. All the flags raised were manually reviewed and re-tested to identify the false positives.

Issues Overview



Project Overview

The KOI token project integrates a ERC20 token with enhanced functionalities through the use of OpenZeppelin contracts.

Core Functionality

At its core, the KOI token is built upon the standard ERC20 contract, ensuring compatibility and interoperability with the broader Ethereum ecosystem.

Token Extensions

The token is extended with the ERC20Burnable contract, allowing users to burn their tokens, effectively reducing the total supply and providing a mechanism for managing inflation or deflation within the token economy. Additionally, the project incorporates the ERC20Permit extension, which facilitates gasless transactions by enabling approvals via off-chain signatures, thus improving user experience by allowing token transfers without the need for holding Ether.

Utility Libraries

The project also leverages a variety of utility libraries such as Math, SignedMath, Strings, ShortStrings, StorageSlot, and cryptographic utilities like ECDSA and MessageHashUtils to ensure safe and efficient operations within the token's smart contracts. These utilities handle key functions such as secure mathematical calculations, efficient string management, and safe storage slot manipulation, all while ensuring the integrity and security of cryptographic operations.

Conclusion

Overall, the KOI token is designed to be a versatile and secure ERC20 token, with advanced features that cater to the needs of end-users within the Ethereum ecosystem.

Scope

In this review, we thoroughly examined the KOI token's smart contracts and associated libraries provided by OpenZeppelin. The review covered the core ERC20 functionality, including how the token handles basic operations such as transfers, allowances, and total supply management. We also focused on the additional features provided by the ERC20Burnable extension, ensuring that the burn functionality was correctly implemented to allow users to reduce the total supply of the token securely.

The ERC20Permit extension was reviewed to confirm that the permit functionality, which allows for gasless approvals via off-chain signatures, is implemented securely. This included an in-depth analysis of the EIP-712 domain separator and the use of the ECDSA library for recovering signer addresses from signatures. We verified that the nonce handling and replay protection mechanisms were correctly integrated to prevent potential attacks.

We also evaluated the utility libraries integrated into the project, such as Math, SignedMath, Strings, ShortStrings, and StorageSlot, to ensure that these libraries were used effectively to handle complex operations, such as arithmetic calculations, string management, and storage handling, without introducing vulnerabilities. The review checked for proper implementation of cryptographic functions and verified that the project adheres to best practices for secure data handling.

Out of Scope

The review did not cover certain aspects of the project that fall outside the direct implementation of the token's functionality. We did not examine the user interface or frontend integrations that might interact with the KOI token. The review did not extend to external smart contracts or decentralized applications (dApps) that may integrate with the KOI token. Additionally, we did not cover the deployment process, network considerations (such as gas optimization strategies during deployment), or the interaction of these contracts with other contracts on the Ethereum blockchain beyond the scope of the provided code. Finally, third-party audits, stress testing, and formal verification processes were not included in this review but are recommended before deploying the token to a live environment.

Methodology

The codebase went through a security review using a filtered code review technique.

- Starting with the reconnaissance phase, a basic understanding was developed.
- The security researchers worked on developing presumptions for the production-ready codebase and the relevant documentation/ white paper provided by the client protocol.
- The security audit moved up to the manual code reviews with the motive of finding logical flaws in the codebase.
- Further complemented with code optimizations, software, and security design patterns implementation, code styles, best practices, and identifying false positives that were detected by automated analysis tools.

Security Review Report

Issues found

Issues	Severity Level	Open	Resolved	Acknowledged
-	Critical	-	-	-
-	High	-	-	-
-	Medium	-	-	-
-	Low	-	-	-
-	Informatory	-	-	-

Summary of Findings

#	Findings	Risk	Status
-	No issues found.	-	-

Findings Overview

No issues found.

Disclaimer

This smart contract audit report (“Report”) is provided by BlockGenesys (“Auditor”) for the benefit of the client (“Client”) and is subject to the following terms and conditions:

The scope of this audit is limited to a review of the smart contract code provided by the Client. The Auditor has conducted a technical analysis of the code to identify potential security vulnerabilities and deviations from best practices. The Auditor has not conducted a formal verification or validation of the functional requirements of the smart contract.

The Auditor has performed the audit to the best of their ability, given the current state of knowledge and technology. However , the Auditor does not warrant that the smart contract
is free from all vulnerabilities or that it will operate as intended in all environments.

The Auditor shall not be liable for any direct, indirect, incidental, special, or consequential damages, including but not limited to loss of data, loss of profits, or any other loss arising from the use or inability to use the smart contract.

This Report is intended solely for the use of the Client and may not be used, reproduced, or distributed to any third party without the prior written consent of the Auditor . The Report does not constitute an endorsement or approval of the smart contract by the Auditor .

The Client is responsible for the deployment and operation of the smart contract. The Client agrees to conduct their own testing and verification of the smart contract to ensure it meets their requirements and to implement any recommended changes identified in the Report.

The Auditor reserves the right to update or revise the Report if new information or vulnerabilities are discovered. The Client is encouraged to keep their smart contract code up-to-date and to periodically review the security of their smart contract.

By engaging the Auditor’s services, the Client acknowledges and agrees to the terms and conditions set forth in this Disclaimer.