QuillAudits

AUDIT REPORT

December, 2024

For

AI AGENT LAYER

Table of Content

Executive Summary

Project name	AiVoiceAgent
Overview	AiVoiceAgent (AIVA) is a project leveraging an ERC20 token to create a digital economy around AI voice services. It mints 1 trillion AIVA tokens at launch, offering standard ERC20 functionalities—transfers, allowances, and balance checks—enhanced with EIP-2612 permit for gasless transactions. Unique features include infinite allowances via Permit2, optimizing user experience in DeFi interactions, and custom hooks for extending token behavior, making it adaptable for future voice-driven applications.
Audit Scope	The scope of this audit was to analyse the AiVoiceAgent for quality, security, and correctness.
Source Code	https://basescan.org/address/0xbdb0e1c40a76c5113a023d685b419b90b01e3d61
Method	Manual Review, Functional Testing, Automated Testing, etc. All the raised flags were manually reviewed and re-tested to identify any false positives.
Timeline	18th December 2024 to 19th December 2024
Blockchain	Base
Fixed In	NA

Checked Vulnerabilities

- ☑ Re-entrancy
- ☑ Timestamp Dependence
- ☑ Gas Limit and Loops
- ☑ DoS with Block Gas Limit
- ☑ Transaction-Ordering Dependence
- ☑ Use of tx.origin
- ☑ Exception Disorder
- ☑ Gasless Send
- ☑ Balance Equality
- ☑ Byte Array
- ☑ Transfer Forwards All Gas
- ☑ Compiler Version Not Fixed
- ☑ Redundant Fallback Function
- ☑ Send Instead of Transfer
- ☑ Style Guide Violation
- ☑ Unchecked External Call
- ☑ Unchecked Math
- ☑ Unsafe Type Inference
- ☑ Implicit Visibility Level
- ☑ Access Management
- ☑ Arbitrary Write to Storage
- ☑ Centralization of Control
- ☑ Ether Theft
- ☑ Improper or Missing Events
- ☑ Logical Issues and Flaws
- ☑ Arithmetic Computations Correctness
- ☑ Race Conditions/Front Running
- ☑ SWC Registry
- ☑ Malicious Libraries
- ☑ Address Hardcoded
- ☑ Divide Before Multiply
- ☑ Integer Overflow/Underflow

- [x] Dangerous Strict Equalities
- [x] Tautology or Contradiction
- [x] Return Values of Low-Level Calls
- [x] Missing Zero Address Validation
- [x] Private Modifier
- [x] Revert/Require Functions
- [x] Multiple Sends
- [x] Using Delegatecall
- [x] Using Suicide
- [x] Using Throw

Techniques and Methods

Throughout the audit of smart contracts, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments, match logic and expected behavior.
- Token distribution and calculations are as per the intended behavior mentioned in the whitepaper.
- Implementation of ERC standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods, and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behavior of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools And Platforms Used For Audit

Remix IDE, Foundry, Solhint, Mythril, Slither, Solidity statistical analysis.

Types of Issues

Types of Severity

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below

High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Issue Status

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

Security vulnerabilities identified that must be resolved and are currently unresolved.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/ impact of the security issue, but are not completely resolved.

No Issues Found

Automated Tests

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of the AiVoiceAgent Token contract. We performed our audit according to the procedure described above.

Contract looks good, No Issues Founds

Disclaimer

QuillAudits Smart contract security audit provides services to help identify and mitigate potential security risks in AiVoiceAgent smart contract. However, it is important to understand that no security audit can guarantee complete protection against all possible security threats. QuillAudits audit reports are based on the information provided to us at the time of the audit, and we cannot guarantee the accuracy or completeness of this information. Additionally, the security landscape is constantly evolving, and new security threats may emerge after the audit has been completed.

Therefore, it is recommended that multiple audits and bug bounty programs be conducted to ensure the ongoing security of AiVoiceAgent smart contract. One audit is not enough to guarantee complete protection against all possible security threats. It is important to implement proper risk management strategies and stay vigilant in monitoring your smart contracts for potential security risks.

QuillAudits cannot be held liable for any security breaches or losses that may occur subsequent to and despite using our audit services. It is the responsibility of the AiVoiceAgent Team to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.

About QuillAudits

QuillAudits is a leading name in Web3 security, offering top-notch solutions to safeguard projects across DeFi, GameFi, NFT gaming, and all blockchain layers. With six years of expertise, we've secured over 1000 projects globally, averting over $30 billion in losses. Our specialists rigorously audit smart contracts and ensure DApp safety on major platforms like Ethereum, BSC, Arbitrum, Algorand, Tron, Polygon, Polkadot, Fantom, NEAR, Solana, and others, guaranteeing your project's security with cutting-edge practices.

6+ Years of Expertise	**1M+** Lines of Code Audited
$30B+ Secured in Digital Assets	**1K+** Projects Secured

Follow Our Journey

AUDIT REPORT

December, 2024

For

AI AGENT LAYER

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com audits@quillaudits.com