



Security Audit

# Ai16ZH

---

January 2, 2025 Asfalia Audit



# Table of Contents

## Summary

### Overview

- Project Summary
- Scope
- Project Overview
- Audit Summary
- Vulnerability Summary
- Project Overview

## [Findings](#)

## Appendix

## Disclaimer

## About

# Summary

This report has been prepared for Ai16ZH to discover issues and vulnerabilities in the source code of the Ai16ZH project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilising Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from Medium to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

# Overview

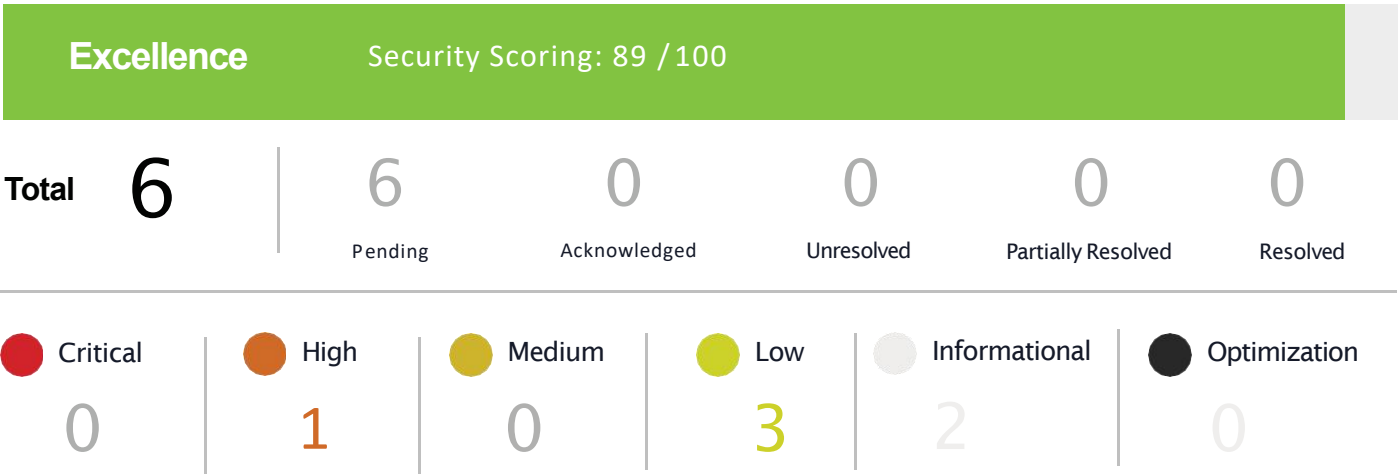
## Project Summary

Project Name	Ai16ZH
Platform	EVM
Chain	ETH
Language	Solidity
Codebase	Files provided
Commit	N/A

## Audit Summary

Delivery Date	10/01/2025
Audit Methodology	Static Analysis, Manual Review

## Vulnerability Summary



## Scope

Repository:	N/A
Technical Documentation:	N/A
Contracts:	<a href="#">Ai16ZH.sol</a>

## Project Overview

Ai16ZH is an innovative project integrating Blockchain and Artificial Intelligence (AI), aiming to establish a decentralized AI computing ecosystem to make AI technology more inclusive, efficient and secure. Through the high-speed blockchain architecture of Solana ecosystem, Ai16ZH is able to provide low-cost, high-throughput AI computing solutions to empower enterprises and individual developers.

## Project Architecture & Fee Models

N/A

## Contract Dependencies

N/A

## Privileged Roles

N/A

# Findings

Total Issues:

6



 Critical	 High	 Medium	 Low	 Informational	 Optimization
0	1	0	3	2	0

ID	Title	Type	Categories	Severity	Status
#1	State Variable Default Visibility	<a href="#">SWC-108</a>	Coding Style	Low	Acknowledged
#2	Unchecked Call Return Value	<a href="#">SWC-104</a>	Coding Style	Low	Acknowledged
#3	Typographical Error	<a href="#">SWC-129</a>	Logical Issue	Low	Acknowledged
#4	Write After Write	<a href="#">Custom</a>	Volatile Code	Informational	Acknowledged
#5	Missing Event	<a href="#">Custom</a>	Coding Style	Informational	Acknowledged
#6	Centralization	<a href="#">Custom</a>	Centralization / Privilege	High	Acknowledged

Ai16ZH.s

## #1 [SWC-108](#) State Variable Default Visibility

Category	Severity	Location	Status
Coding Style	Low	Line 342, 343	Acknowledged

### Description

The visibility is not defined for various state variables.

### Recommendation

It is best practice to set the visibility of state variables explicitly. The default visibility for state variables is internal. Other possible visibility settings are public and private. Recommend using public.

### Alleviation

N/A

## #2 [SWC-104](#) Unchecked Call Return Value

Category	Severity	Location	Status
Coding Style	Low	Line 733	Acknowledged

### Description

Return value for the addLiquidityETH function call is not checked.

### Recommendation

Should check the return value of addLiquidityETH is true to ensure liquidity is being added correctly.

### Alleviation

N/A

**#3 SWC-129 Typographical Error**

Category	Severity	Location	Status
Volatile Code	Low	Line 769, 770	Acknowledged

**Description**

In Solidity multiplication should come before division to avoid rounding errors.

**Recommendation**

Restructure the mathematical operation.

**Alleviation**

N/A

**#4 Custom Write After Write**

Category	Severity	Location	Status
Coding Style	Informational	Line 783, 785	Acknowledged

**Description**

The bool `success` is declared and then defined twice for two calls.

**Recommendation**

Bools with different defined names should be used for each different check on a call performing correctly.

**Alleviation**

N/A



## #5 Custom Missing Event

Category	Severity	Location	Status
Coding Style	Informational	Line 517-521, 560-567, 569-576	Acknowledged

### Description

Functions missing an event for functions with significant contract change impact. Information about changed variables should be emitted as an event for third parties to more easily track.

### Recommendation

Add events that emit the changed variable information, for `updateSwapTokensAtAmount`, `updateBuyFees`, `updateSellFees`.

### Alleviation

N/A

## Centralization

Category	Status
Centralization / Privilege	Acknowledged

### Description

The contract uses the OpenZeppelin Ownership contract format to allow the developers to call several functions that affect how the contract functions. These functions allow the developers to:

- Clear stuck ETH or foreign tokens on the contract
- Change the buy or sell fees of the token (with a maximum of 10%)
- Define the automated market maker pair of the token
- Airdrop tokens
- Change the amount of tokens needed to accumulate in fees before it is sold (minimum 0.001% of supply, maximum 0.1% of supply)
- Enable trading and remove initial limits once token is stable
- Define addresses as `boughtEarly` (bots)

## #5 Custom Missing Event

Category	Severity	Location	Status
Centralization / Privilege	High	Line 483-485 & 487-490	Acknowledged

### Description

Functions `manageBoughtEarly` and `massManageBoughtEarly` allow for the owner role to manually declare an address as `true` or `false` on the `boughtEarly` array. This makes an address only able to transfer tokens to the owner or burn address. Intended purpose seems to be to manually flag any bots that are not caught by the contracts anti-bot code.

### Recommendation

Recommend removing this function, or alternatively adding a timed safeguard so that it may only be used soon after trading goes live and once all bots have been handled.

### Alleviation

N/A

# Appendix

## Finding Categories

### Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

### Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

### Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

### Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.

### Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

### Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

### Block Timestamp

Be aware that the timestamp of the block can be manipulated by a miner.

# Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement.

This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without Asfalia's prior written consent in each instance. This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Asfalia to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-freenature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. Asfalia's position is that each company and individual are responsible for their own due diligence and continuous security. Asfalia's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by Asfalia is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

Project is potentially vulnerable to 3rd party failures of service - namely in the form of APIs providing the price for the currencies used by the project. Project could become at risk if these APIs provided incorrect pricing.

Audit does not claim to address any off-chain functions utilized by the project.



The firm was started by a team with over ten years of network security experience to become a global force. Our goal is to make the blockchain ecosystem as secure as possible for everyone.

With over 30 years of combined experience in the DeFi space, our team is highly dedicated to delivering a product that is as streamlined and secure as possible. Our mission is to set a new standard for security in the auditing sector, while increasing accessibility to top tier audits for all projects in the crypto space. Our dedication and passion to continuously improve the DeFi space is second to none.