AstraSec

AI SweatShop Security Audit Report

December 30, 2024

Contents

1 Introduction

1.1 About AI SweatShop

AI SweatShop is a decentralized platform for AI token management, featuring a dynamic market cap-based fee system, automated LP token vesting, Chainlink-powered price feeds, and Uniswap V2 liquidity integration, enabling efficient and transparent tokenomics for blockchain projects.

1.2 Source Code

The following source code was reviewed during the audit:

- https://github.com/open-overlord/ass-contracts.git
- CommitID: 9988140

And this is the final version representing all fixes implemented for the issues identified in the audit:

- https://github.com/open-overlord/ass-contracts.git
- CommitID: b00b60a

2 Overall Assessment

This report has been compiled to identify issues and vulnerabilities within the AI SweatShop project. Throughout this audit, we identified a total of 2 issues spanning various severity levels. By employing auxiliary tool techniques to supplement our thorough manual code review, we have discovered the following findings. It is important to clarify that this audit does not cover economic aspects, and the correctness or reasoning behind the underlying invariants is outside the scope of this assessment.

Severity	Count	Acknowledged	Won't Do	Addressed
Critical	–	–	–	–
High	–	–	–	–
Medium	1	1	–	–
Low	1	–	–	1
Informational	–	–	–	–
Undetermined	–	–	–	–

3 Vulnerability Summary

3.1 Overview

Click on an issue to jump to it, or scroll down to see them all.

3.2 Security Level Reference

In web3 smart contract audits, vulnerabilities are typically classified into different severity levels based on the potential impact they can have on the security and functionality of the contract. Here are the definitions for critical-severity, high-severity, medium-severity, and low-severity vulnerabilities:

Severity	Acknowledged
C-X (Critical)	A severe security flaw with immediate and significant negative consequences. It poses high risks, such as unauthorized access, financial losses, or complete disruption of functionality. Requires immediate attention and remediation.
H-X (High)	Significant security issues that can lead to substantial risks. Although not as severe as critical vulnerabilities, they can still result in unauthorized access, manipulation of contract state, or financial losses. Prompt remediation is necessary.
M-X (Medium)	Moderately impactful security weaknesses that require attention and re-mediation. They may lead to limited unauthorized access, minor financial losses, or potential disruptions to functionality.
L-X (Low)	Minor security issues with limited impact. While they may not pose significant risks, it is still recommended to address them to maintain a robust and secure smart contract.
I-X (Informational)	Warnings and things to keep in mind when operating the protocol. No immediate action required.
U-X (Undetermined)	Identified security flaw requiring further investigation. Severity and impact need to be determined. Additional assessment and analysis are necessary.

3.3 Vulnerability Details

3.3.1 [M-1] Possible Manipulation of Market Cap in _getMarketCap()

TARGET	CATEGORY	IMPACT	LIKELIHOOD	STATUS
AgentToken.sol	Business Logic	Medium	Medium	Acknowledged

In the AI SweatShop project, the LP tokens generated from the initial liquidity are locked in a Vesting contract. The release timing of these locked LP tokens depends on the market cap of the AI token. Additionally, whether the AI token charges fees during buying and selling also depends on its market cap. During our review of the market cap calculation for the AI token, we identified a risk of manipulation. Specifically, the market cap calculation for the AI token directly depends on reserve0 and reserve1 from the liquidity pool. These reserves can be manipulated through swap operations, thereby affecting the calculated market cap. As a result, by intentionally manipulating the reserves in the liquidity pool, the release timing of the locked LP tokens and the enabling or disabling of the AI token's fee mechanism can both be manipulated. If the market cap is around $2M, manipulating the enabling and disabling of the AI token's fee mechanism may create various arbitrage opportunities, causing potential asset losses for users.

ass-contracts-main - AgentToken.sol

```
152 function _getMarketCap() internal view returns (uint256 marketCap) {
153     ...
154     (uint256 reserve0, uint256 reserve1,) = IUniswapV2Pair(pair).getReserves();
155     if (reserve0 == 0 || reserve1 == 0) return 0;
156     // calculate token price in ETH
157     uint256 tokenPriceInEth;
158     if (IUniswapV2Pair(pair).token0() == address(this)) {
159         tokenPriceInEth = (reserve1 * 1 ether) / reserve0;
160     } else {
161         tokenPriceInEth = (reserve0 * 1 ether) / reserve1;
162     }
163     ...
164     // market cap: token price * total supply * ETH price
165     marketCap = tokenPriceInEth * MAX_SUPPLY * ethPrice / 1 ether / 1 ether;
166 }
```

Remediation Use the TWAP (Time-Weighted Average Price) of the AI token instead of the instantaneous price to calculate the market cap.

3.3.2 [L-1] Improved Implementation Logic in Vesting::lock()

TARGET	CATEGORY	IMPACT	LIKELIHOOD	STATUS
Vesting.sol	Business Logic	Low	Low	Addressed

In the Vesting contract, the lock() function retrieves the factory parameter from the token address provided by the user. It then checks whether the retrieved factory address matches the factory address of Uniswap V2 to determine if the token contract is a legitimate Uniswap V2 pair contract. However, if the user provides a malicious token contract, this check can be bypassed, allowing malicious tokens that are not Uniswap V2 LPs to be locked in the vesting contract. Note that the startRelease() function in the same contract shares the same issue.

ass-contracts-main - Vesting.sol

```
50  /// @inheritdoc IVesting
51  function lock(address token, uint256 amount, address beneficiary) external override {
52      if (beneficiary == address(0)) revert ZeroAddress();
53      if (amount == 0) revert ZeroAmount();
54      // tokens can only be locked once
55      if (_vestings[token].beneficiary != address(0)) revert TokenAlreadyLocked();
56      // check if the token is uniswap pair
57      if (!_isUniswapV2Pair(token)) revert TokenNotFromUniswapV2Pair(token);
58
59      // save vesting info
60      _vestings[token] =
61          VestingInfo({beneficiary: beneficiary, amount: amount, startTime: 0, released: 0, lastReleasePeriod: 0});
62
63      IERC20(token).safeTransferFrom(msg.sender, address(this), amount);
64
65      emit TokenLocked(token, beneficiary, amount);
66  }
```

Remediation Add restrictions to the calls of the lock() and startRelease() functions, allowing only the AISweatShop contract to invoke them.

4 Appendix

4.1 About AstraSec

AstraSec is a blockchain security company that serves to provide high-quality auditing services for blockchain-based protocols. With a team of blockchain specialists, AstraSec maintains a strong commitment to excellence and client satisfaction. The audit team members have extensive audit experience for various famous DeFi projects. AstraSec's comprehensive approach and deep blockchain understanding make it a trusted partner for the clients.

4.2 Disclaimer

The information provided in this audit report is for reference only and does not constitute any legal, financial, or investment advice. Any views, suggestions, or conclusions in the audit report are based on the limited information and conditions obtained during the audit process and may be subject to unknown risks and uncertainties. While we make every effort to ensure the accuracy and completeness of the audit report, we are not responsible for any errors or omissions in the report.

We recommend users to carefully consider the information in the audit report based on their own independent judgment and professional advice before making any decisions. We are not responsible for the consequences of the use of the audit report, including but not limited to any losses or damages resulting from reliance on the audit report.

This audit report is for reference only and should not be considered a substitute for legal documents or contracts.

4.3 Contact

Phone	+86 156 0639 2692
Email	contact@astrasec.ai
Twitter	https://twitter.com/AstraSecAI