

# Aegis AI

Pioneering Blockchain Security with  
AI-Enabled Audit Solutions.



# Table of Contents

- 03. About Aegis AI
- 04. Introduction
- 05. Project Overview
- 06. Social Media
- 07. Audit Summary
- 08. Vulnerability and Risk Level
- 09. Auditing Strategy and Techniques
- 11. Overall Security
- 14. Ownership
- 20. External / Public functions
- 22. Capabilities
- 23. Inheritance Graph
- 25. Audit Results
- 26. Files Overview
- 27. Conclusion
- 29. Glossary

# About **Aegis AI**

Aegis AI is a revolutionary tool designed to bring accessibility, transparency, and trust to the world of blockchain technology. With the increasing use of smart contracts in various industries, the need for efficient and user-friendly auditing tools has never been more critical. Aegis AI is the solution that bridges the gap between complex smart contract code and non-technical users, making it easy for anyone to ensure the security and reliability of their digital assets and transactions.

- Run quick audits from dApp using AI
- Generate detailed audit reports
- Monitor of smart contracts and protocols in real time.
- Automated Penetration Testing.



# Introduction

Aegis AI offers an advanced AI-driven solution for smart contract auditing, designed to enhance the security of blockchain contracts with ease. Our platform caters to users of all skill levels, enabling thorough vulnerability assessments without requiring extensive coding knowledge. This user-friendly tool simplifies the auditing process, efficiently detecting potential security risks and malicious code within smart contracts.

By addressing critical security concerns, Aegis AI plays a vital role in fostering trust and growth in blockchain technology. Our solution is integral to promoting wider adoption of smart contracts, ensuring their reliability and integrity across various blockchain applications.



## Project Overview



Aegis  
AI

Project Name	Intellix
Symbol	ITX
Address	0x23D894fb4A0f551F2f923fc85e09819D1F3894b2
Type	ERC-20
Decimals	18
Total Supply	1,000,000,000
Market Cap	4413000
Exchange Rate	0.004413
Holders	1,297

# Social Media

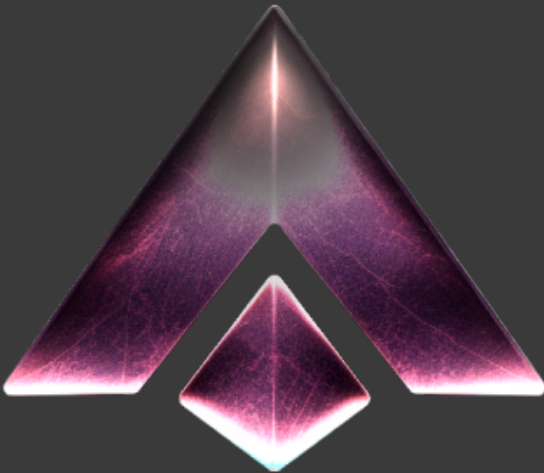


**Audit Summary**

Version	Delivery Date	Changelog
1.0	January 09, 2024	Layout project Automated / Manual Security Testing Summary

**Note**

This Audit report consists of a security analysis of the Intellix smart contract.



# Vulnerability and Risk Level

Risk assessment gauges the likelihood and impact of potential threats exploiting vulnerabilities. It is quantified using **CVSS version 3.0** standards, providing a clear metric for organizational or system security evaluation.

Severity Level	CVSS Score	Description	Recommended Action
Critical	9-10	Severe threat with potential for major losses or complete failure.	Immediate action required to mitigate and resolve.
High	7-8.9	Can seriously compromise functionality and security, allowing potential exploitation.	Address promptly to prevent exploitation.
Medium	4-6.9	Impact certain aspects, potentially leading to unintended behaviors.	Correct within a reasonable timeframe to maintain integrity.
Low	2-3.9	Less likely to significantly impact performance but should be addressed.	Consider remediation, may accept risk based on context.
Information	0-1.9	Highlights areas for improvement or optimization, no security risk.	Review for potential enhancements, no immediate action.



## **Auditing Strategy and Techniques**

In our review process, we leverage advanced, finetuned Large Language Models (LLMs) alongside sophisticated LLM agent mechanisms. This approach ensures comprehensive examination of the smart contract repository, targeting security vulnerabilities, code integrity, and adherence to the latest standards and best practices.

Our review blends the precision of machine learning models with human oversight. The finetuned LLMs efficiently parse and analyze every file, providing in-depth insights and faster results, which are then meticulously validated by our team of experts for accuracy and relevance.

## **Methodology**

The audit of the smart contract Intellix was conducted using a systematic and risk-based approach. Emphasis was placed on essential aspects such as security, code quality, compliance, gas efficiency, and overall functionality.

### **Security Assessment**

Our approach integrates advanced AI-driven techniques, primarily leveraging Large Language Models (LLMs) and GPT agents. This blend of AI tools provides an in-depth analysis, identifying and evaluating potential security vulnerabilities. Alongside this, we conduct targeted manual reviews to validate and contextualize the AI-generated insights, ensuring comprehensive and accurate security assessments.

### **Code Quality Evaluation:**

LLMs played a key role in examining the code's quality, focusing on readability and maintainability. The AI analysis, combined with our expert review, ensured compliance with smart contract development standards.

### **Compliance Review:**

Our audit included an LLM-assisted compliance check against industry standards like ERC-20 and ERC-721. This process pinpointed deviations, providing a basis for our detailed compliance recommendations.

### **Gas Efficiency Analysis:**

AI tools evaluated the contract's gas consumption, offering insights into its execution efficiency. These findings were enhanced by expert analysis to suggest practical optimization strategies.


### **Functionality and Logic Verification:**

The audit applied LLMs to validate the contract's functionality and logic, ensuring its operations matched the intended design. This automated check was complemented by thorough manual testing.

## Overall Security


### Honeypot

Honeypots are smart contracts that appear to have an obvious flaw in their design, which allows an arbitrary user to drain ether (Ethereum's cryptocurrency) from the contract, given that the user transfers a priori a certain amount of ether to the contract.

Is it a honeypot?  The contract is not a Honey Pot	
<b>Description</b>	Owner cannot drain your wallet through honeypot

### Antiwhale

Certain features adopted to prevent large holders (aka whales) from exerting excessive influence or engaging in manipulative behaviors within the token ecosystem. Some examples are setting maximum transaction limits, imposing penalties for transactions exceeding some specific threshold, imposing a more equitable distribution of tokens

Can whales dump?  The contract is Anti Whale	
<b>Description</b>	Whales can't dump

## Listing

Listings on multiple decentralized exchanges (DEX) with good amount of liquidity is a good sign

Is it on a dex?		✓ The contract is listed	
<b>Description</b>		You can swap tokens on dex	

## Opensource

Open source contract is contract with source code that anyone can inspect, modify, and enhance.

Is code available?		✓ The contract is Open Source	
<b>Description</b>		Contract code can be reviewed and audited by anyone	

## Proxy

Proxy contract is a contract that delegates calls to another contract. It is a contract that has a fallback function that calls another contract. If the proxy contract is well-designed, secure, and serves a legitimate purpose (such as upgradability or modularity), it may not raise concerns. However, if the proxy introduces vulnerabilities, lacks transparency, or is used in a way that compromises the security of the token, it could be flagged during a thorough audit.

Is it a proxy?

✓ The contract is not a Proxy contract

**Description**

This is a full contract

## Ownership

Is ownership renounced?	
✗ Contract has an owner	
<b>Description</b>	The owner has not renounced the ownership that means that the owner retains control over the contract's operations, including the ability to execute functions that may impact the contract's users or stakeholders. This can lead to potential issues.
<b>Comments</b>	<b>Centralization</b> The owner has significant control over contract's operations.

### Note

If the contract is not deployed then we would consider the ownership to be not renounced. Moreover, if there are no ownership functionalities, ownership is automatically considered renounced.

## Ownership Privileges

These functions can be dangerous. Please note that abuse can lead to financial loss. We have a guide where you can learn more about these Functions.

### Minting Privileges

Minting is the process of creating new tokens. This is usually done by the contract owner, and the newly minted tokens are added to the owner's balance. Minting is usually done to increase the total supply of a cryptocurrency or token.

Can mint new tokens?

✓ The owner cannot mint new tokens

#### Description

The owner cannot mint new tokens

## Modify Balance

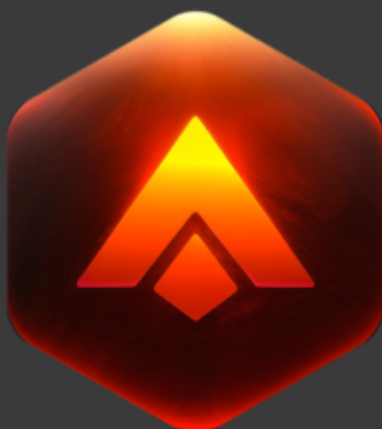
If the owner can modify the token holders balance, it can be used to steal tokens from the token holders. The owner can also burn tokens without any allowance.

Can modify  
balances?

✓ The owner cannot modify token balance

### Description

The owner is not able to modify  
token balance without any  
allowance





## Blacklist addresses

Blacklisting addresses in smart contracts is the process of adding a certain address to a blacklist, effectively preventing them from accessing or participating in certain functionalities or transactions within the contract. This can be useful in preventing fraudulent or malicious activities, such as hacking attempts or money laundering.

Can blacklist  
addresses?

☒ Owner cannot blacklist addresses


### Description

The contract owner cannot blacklist  
addresses



## Fees and tax

In some smart contracts, the owner or creator of the contract can set fees for certain actions or operations within the contract. These fees can be used to cover the cost of running the contract, such as paying for gas fees or compensating the contract's owner for their time and effort in developing and maintaining the contract.

Is there a tax?	
 Contract takes a tax	
Description	<p>There is a tax to the contract owner when you buy and sell the token</p> <p><b>Tax on buy:</b> 5% <b>Tax on sell:</b> 5%</p>



## Self Destruct

In a smart contract, the selfdestruct feature refers to a specific function that, when executed, destroys the contract and removes it from the blockchain. This action renders the contract inoperative and ends all its functions. When a contract is self-destructed, any remaining balance in the contract is sent to a designated address, and the contract's code and storage are removed from the state of the blockchain.

Can self  
destruct?

✓ Contract cannot self destruct

### Description

The smart contract does not include a self-destruct feature.



## External / Public functions

External/public functions are functions that can be called from outside of a contract, i.e., they can be accessed by other contracts or external accounts on the blockchain. These functions are specified using the function declaration's external or public visibility modifier.

## State variables

State variables are variables that are stored on the blockchain as part of the contract's state. They are declared at the contract level and can be accessed and modified by any function within the contract. State variables can be denoted with a visibility modifier, such as public, private, or internal, which determines the access level of the variable.

### Components

External	Internal	Private	Pure
20	11	1	0

**Exposed Functions**

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public state variables are not included.

Public	Payable
20	0

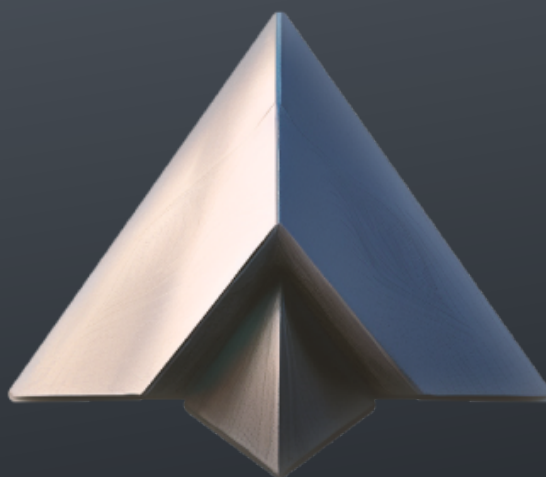
External	Internal	Private	Pure	View
20	11	1	0	36

**State Variables**

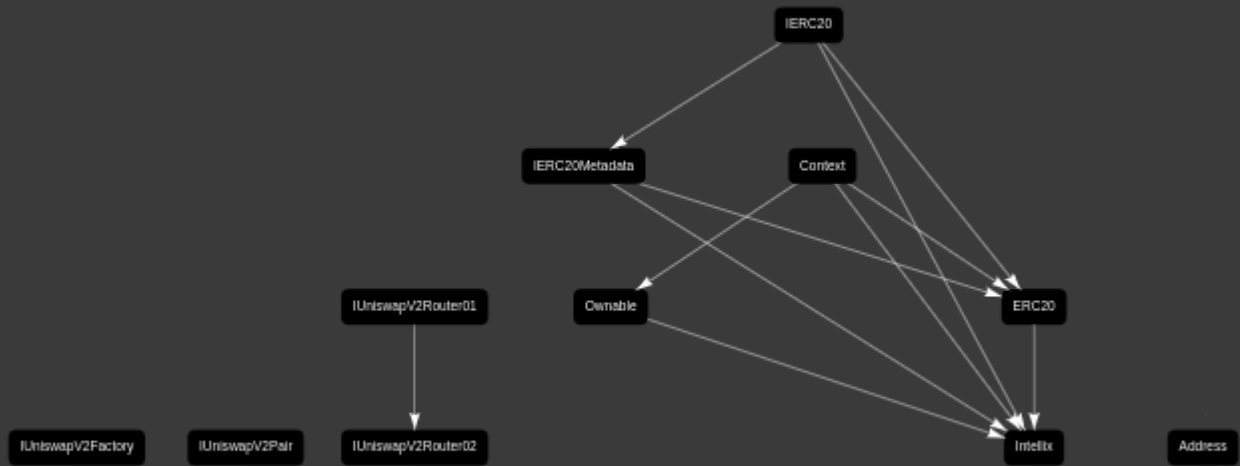
Total	Public
113	36

## Capabilities

Aegis Version observed	Transfers ETH	Can Receive Funds	Uses Assembly	Delegate Call
$\geq 0.6.0$ $< 0.9.0$	Yes	Yes	Yes	Yes



## Inheritance Graph



## Centralization Privilege

Centralization can arise when one or more parties have privileged access or control over the contract's functionality, data, or decision-making. This can occur, for example, if the contract is controlled by a single entity or if certain participants have special permissions or abilities that others do not. In the project, there are authorities that have access to the following functions:

Contract	Privileges
IUniswapV2Factory	createPair, setFeeTo, setFeeToSetter
IUniswapV2Pair	approve, transfer, transferFrom, permit, mint, burn, swap, skim, sync, initialize
IUniswapV2Router01	addLiquidity, removeLiquidity, removeLiquidityETH, removeLiquidityWithPermit, removeLiquidityETHWithPermit, swapExactTokensForTokens, swapTokensForExactTokens, swapTokensForExactETH, swapExactTokensForETH
IUniswapV2Router02	removeLiquidityETHSupportingFeeOnTransferTokens, removeLiquidityETHWithPermitSupportingFeeOnTransferTokens, swapExactTokensForTokensSupportingFeeOnTransferTokens, swapExactTokensForETHSupportingFeeOnTransferTokens
IERC20	transfer, approve, transferFrom
IERC20Metadata	N/A
Address	N/A
Context	N/A
Ownable	renounceOwnership, transferOwnership
ERC20	transfer, approve, transferFrom, increaseAllowance, decreaseAllowance
Intellix	claimStuckTokens, setTierSettings, excludeFromFees, updateFees, changeITXVaultWallet, changeITXRevtWallet, enableTrading, setSwapEnabled, setSwapTokensAtAmount, swapAndSendITX, setEnableMaxWalletLimit, setMaxWalletAmount, excludeFromMaxWallet



**Audit Results****#AEG-1 Reentrancy**

FILE	Severity
Intellix.sol	INFO

**Description** - The function 'swapAndSendITX' interacts with an external contract (UniswapV2Router) to swap tokens for ETH and then sends ETH to two different wallets. The external call to send ETH is not protected against reentrancy attacks. An attacker could exploit this by creating a fallback function in the receiving wallet contract that calls back into the smart contract, potentially draining its funds in a loop.

**#AEG-2 Arbitrary Swap Amount**

FILE	Severity
Intellix.sol	INFO

**Description** - The owner can set any 'swapTokensAtAmount' which could be exploited by setting an extremely high value that never triggers the swap, thus disabling the fee distribution mechanism. Alternatively, setting it very low could lead to excessive swaps on every transaction, causing significant slippage and impacting the token's price.

**Audit Results****#AEG-3 Disproportionate Fees**

FILE	Severity
Intellix.sol	INFO

**Description** - The function '\_transfer' applies fees of 99% for the first block after trading is enabled, which is excessively high and could be seen as a 'honeypot' to trap users' funds. Additionally, a high sell fee of 20% is applied for the first two hours, which could be used to prevent selling and manipulate the token price.

## Files Overview

The Intellix team provided us with the files that should be tested in the security assessment. This audit covered the following files listed below with an SHA-1 Hash.

File Name

Intellix.sol

## Imported Packages

Used code from other Frameworks/Smart Contracts (direct imports).

## Note for Investors:

We only audited a token contract for Intellix. However, If the project has other contracts (for example, a Presale, staking contract etc), and they were not provided to us in the audit scope, then we cannot comment on its security and are not responsible for it in any way.

No external libraries used.

## Source

**language:** solidity

**version:** 0.8.19+commit.7dd6d404

**verified at:** January 4, 2024 at 02:47:25 PM (UTC+0)

# Conclusion

The conclusion of the audit for Intellix (ITX), has delivered promising results, bolstering confidence in the robustness and security of its codebase. No significant risks have been identified, an aspect which is testament to its solid infrastructure. This includes a clean issue slate with no high, medium or low severity concerns. This positive assessment underscores the soundness of this digital currency and its potential to operate with ample stability and reliability in the volatile crypto market. The audit shines light on various aspects of ITX. On one hand, its sell tax rate stands at 6.3% and the buy tax rate at 5%, a touch above average. However, considering operational expenses and potential for reinvestment, these rates may not be a major concern for seasoned investors. They may, in fact, secure the value of this currency over time. Moreover, it is noteworthy that ITX is not a honeypot, a characteristic which bodes well for its reliability. Intellix also features anti-whale measures, a prudent move that should prevent undue manipulation of the currency's value by large players, thus safeguarding the interests of smaller investors. The fact that ITX is not mintable ensures the preservation of its scarcity, a significant factor when considering the future appreciation of its value. With no blacklisting, the token allows unrestricted trading - a reflection of its commitment to providing an open and transparent marketplace. As substantiated by the audit, the currency is listed on a decentralized exchange (dex), a typical procedure which opens it up to investors worldwide. An indication of its robust adoption is its already impressive base of 1289 holders, facilitated by the fact that it has more than three liquidity providers, a positive indication of user interest. To conclude, the audit paints a favorable picture of Intellix (ITX) as a stable, reliable digital token with solid market interest and ample safeguards in place. It is a token that upholds inclusivity, transparency, and a fair play culture, making it a strong contender in the digital currency space.

## Conclusion Overview

Overview	Notes	Result
<b>Honeypot</b>	The contract owner can drain the funds from the contract	✓ False
<b>Anti whale check</b>	Features preventing whales from manipulating the Token	✓ True
<b>Opensource</b>	The code of the contract is public	✓ True
<b>Ownership renounced</b>	Contract owner has renounced ownership	✗ False
<b>Buy tax</b>	Fees incurred when buying the token	✓
<b>Sell tax</b>	Fees incurred when selling the token	✓
<b>High Severity Issues</b>	Number of High severity issues	0
<b>Medium Severity Issues</b>	Number of Medium severity issues	0
<b>Mintable</b>	Can mint new tokens	✓ False
<b>Blacklist</b>	Owner can blacklist users	✓ False
<b>Holders</b>	Total wallets holding the token	1289
<b>LP holder</b>	Number of liquidity providers	3

# Glossary

## 1. Honeypot:

A honeypot refers to a deceptive contract that lures investors by appearing lucrative or profitable. These contracts typically allow users to easily purchase tokens, but selling them is restricted or impossible. This tactic is used to trap funds, misleading investors who are unable to withdraw their investments.

## 2. Blacklist:

A blacklist refers to a mechanism that enables the contract owner to restrict certain addresses from buying or selling the token. This feature is often implemented to block suspected bots or malicious actors from manipulating the token's market. However, it can also be used to unfairly prevent legitimate users from selling their tokens, posing a risk to token holder rights.

## 3. Ownership Privileges:

Ownership privileges refer to the exclusive rights and controls a contract owner possesses. These can include altering critical contract parameters, managing listings on decentralized exchanges, and updating contract logic. While revoking ownership can enhance trust among users by making the contract immutable, maintaining ownership is crucial for larger projects that require ongoing management and adaptability to evolving blockchain ecosystems.

#### **4. Automated Penetration Testing:**

Automated Penetration Testing is a cybersecurity practice that employs automated tools and technologies to identify and exploit vulnerabilities in computer systems, networks, or applications. It aims to simulate potential cyberattacks to assess the security posture and discover weaknesses in order to enhance overall defense against malicious activities.

#### **5. LLM:**

A large language model (LLM) is a type of artificial intelligence (AI) algorithm that uses deep learning techniques and massively large data sets to understand, summarize, generate and predict new content.

#### **6. CVSS:**

CVSS stands for the Common Vulnerability Scoring System. It's a way to evaluate and rank reported vulnerabilities in a standardized and repeatable way.

#### **7. EOA:**

Externally Owned Accounts (EOAs) are the most common type of blockchain account that gives us direct control. These accounts are created using private keys. The associated key gives you a unique signature and access to the blockchain. You can use it to send and receive transactions and interact with applications.