

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 Applied

11. Overall Security

14. Ownership

15. Ownership Privileges

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 is an AI-powered smart contract auditing tool that empowers end users with the ability to assess and enhance the security of their smart contracts, even without any coding knowledge. Aegis AI offers a seamless and intuitive interface, allowing users to audit smart contracts with a few clicks, instantly identifying vulnerabilities and malicious elements.

Addressing these issues is essential to realizing the full potential of blockchain technology and smart contracts. A solution that empowers users, regardless of their technical background, to easily audit and secure their smart contracts is not only desirable but imperative for the continued adoption and trust in blockchain ecosystems.



Project Overview



Aegis AI

Project Name	ChainLink Token
Symbol	LINK
Address	0x514910771AF9Ca656af840dff83E8264EcF986CA
Type	ERC-20
Decimals	18
Total Supply	1,00,00,00,000
Market Cap	8,626,199,075.03
Exchange Rate	15.51
Holders	0

Social Media



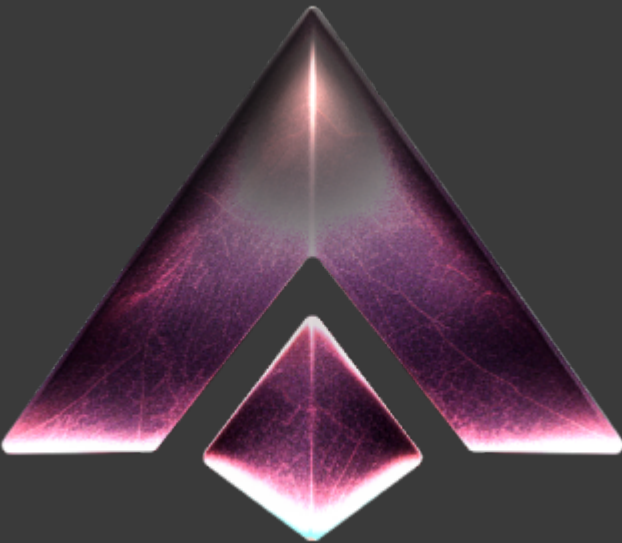
Audit Summary

Version	Delivery Date	Changelog
1.0	December 15, 2023	<ul style="list-style-type: none">Layout projectAutomated / Manual Security TestingSummary

Note

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

This analysis did not include functional testing (or unit testing) of the contract’s logic



Vulnerability and Risk Level

Risk represents the probability that a certain source threat will exploit vulnerability and the impact of that event on the organization or system.

The risk Level is computed based on CVSS version 3.0.

Choose your plan	Value	Vulnerability	Risk (Required Action)
Critical	9-10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level
High	7-8.9	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level
Medium	4-6.9	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Caution advised
Low	2-3.9	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Awareness and monitoring
Informational	0-1.9	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Awareness and monitoring

Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to check the repository for security-related issues, code quality, and compliance with specifications and best practices. To this end, our team of experienced pen-testers and smart contract developers reviewed the code line by line and documented any issues discovered.

We check every file manually. We use automated tools only so that they help us achieve faster and better results.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:

- Reviewing the specifications, sources, and instructions provided to SolidProof to ensure we understand the size, scope, and functionality of the smart contract.
- Manual review of the code, i.e., reading the source code line by line to identify potential vulnerabilities.
- Comparison to the specification, i.e. Verifying that the code does what is described in the specifications, sources, and instructions provided to SolidProof.

2. Testing and automated analysis that includes the following:

- Test coverage analysis determines whether test cases cover code and how much code is executed when those test cases are Executed.
- Symbolic execution, which is analysing a program to determine what inputs cause each part of a program to execute.


3. Review best practices, i.e., review smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on best practices, recommendations, and research from industry and academia.

4. Concrete, itemized and actionable recommendations to help you secure your smart contracts.

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	
Description	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 as more equitable distribution of tokens

Can whales dump?  The contract is not Anti Whale	
Description	Whales might 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	
Description	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	
Description	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

The ownership is renounced

✔ Contract does not have an owner

Description	The owner has renounced the ownership that means that the owner has no control over the contract’s operations, including the ability to execute functions that may impact the contract’s users or stakeholders.
Comments	N/A

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.

Contract owner cannot mint new tokens

✅ The owner cannot mint new tokens

Description

The owner cannot mint new tokens

Burning Tokens

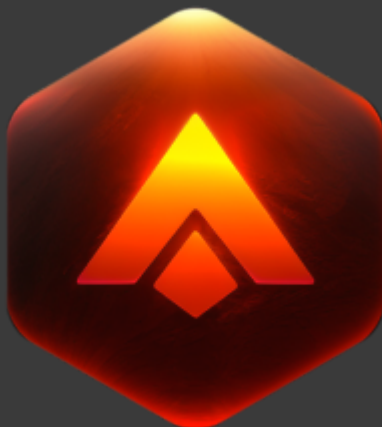
Burning tokens is the process of permanently destroying a certain number of tokens, reducing the total supply of a cryptocurrency or token. This is usually done to increase the value of the remaining tokens, as the reduced supply can create scarcity and potentially drive up demand.

Contract owner cannot burn tokens

☒ The owner cannot burn tokens

Description

The owner is not able burn tokens without any allowance



Blacklist addresses

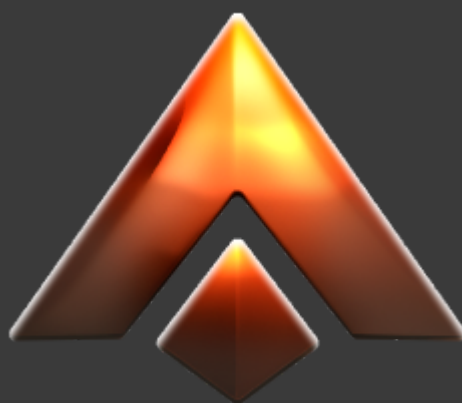
Blacklisting addresses in smart contracts is the process of adding a certain address to a blacklist, effectively preventing them from accessing or participating in certain functionalities or transactions within the contract. This can be useful in preventing fraudulent or malicious activities, such as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

Contract owner cannot blacklist addresses

☒ The owner cannot blacklist addresses

Description

The 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.

There is a buy tax of 0.00%

There is a sell tax of 0.00%


Description

There is a tax to the contract owner when you buy or sell the token



Lock User Funds

In a smart contract, locking refers to the process of restricting access to certain tokens or assets for a specified period of time. When tokens or assets are locked in a smart contract, they cannot be transferred or used until the lock-up period has expired or certain conditions have been met.

Owner Cannot lock the contract	
 The owner cannot lock the contract	
Description	The owner is not able to lock the contract by any functions or updating any variables.

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 defined with a visibility modifier, such as public, private, or internal, which determines the access level of the variable.

Components

External	Internal	Private	Pure
20	4	2	0

Exposed Functions

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

Public	Payable
20	0

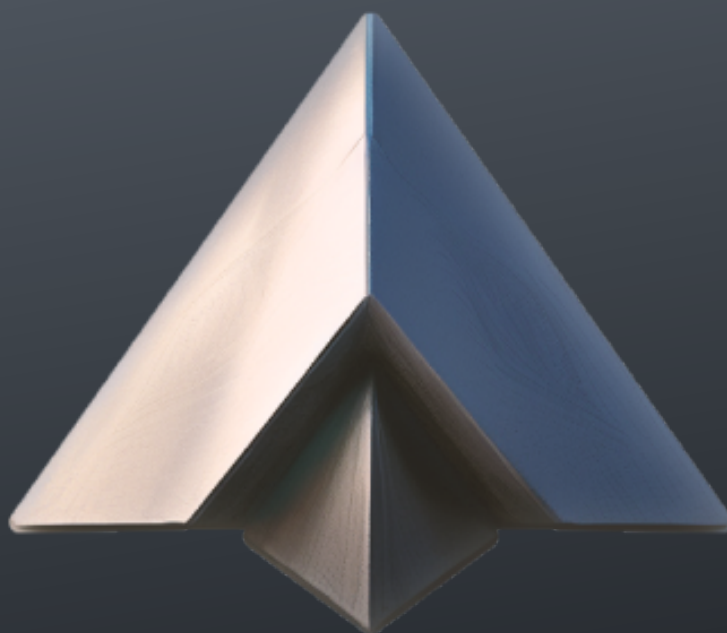
External	Internal	Private	Pure	View
20	4	2	0	6

StateVariables

Total	Public
26	6

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 Privileges

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
SafeMath	N/A
ERC20Basic	transfer
ERC20	transferFrom, approve
ERC677	transferAndCall
ERC677Receiver	onTokenTransfer
BasicToken	transfer
StandardToken	transferFrom, approve, increaseApproval, decreaseApproval
ERC677Token	transferAndCall, contractFallback, isContract
LinkToken	transferAndCall, transfer, approve, transferFrom

Audit Results**#AEG-1 Missing Implementation**

FILE	Severity
ChainLink Token.sol	MEDIUM

Description - The division function is incomplete and lacks the actual implementation code for division, which could lead to undefined behavior if this function is used in the contract.

#AEG-2 Outdated Compiler Version

FILE	Severity
ChainLink Token.sol	HIGH

Description - The smart contract is using an outdated version of Solidity (0.4.16). This version lacks many security features and optimizations that are present in newer versions of the Solidity compiler. It is recommended to use the latest version to benefit from improved security measures and language features.

Files Overview

The ChainLink Token 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.

No files provided.

Note

Files with a different hash value than in this table have been modified after the security check, either intentionally or unintentionally. A different hash value may (but need not) be an indication of a changed state or potential vulnerability that was not the subject of this scan.

Imported Packages

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

Note for Investors:

We only audited a token contract for ChainLink Token. 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: v0.4.16+commit.d7661dd9

verified at: 2019-04-17T21:44:56.971525Z

Conclusion

The audit report outlines Immutable X token, designated by IMX, as apart from honeypot and blacklist risks, lending it creditworthiness. Exhibiting a diverse distribution of external, internal, private, and view functions, there are some high & medium-severity issues, warranting attention. Immutability is slightly tarnished with a lack of antiwhale measures which poses an investor risk. An exceptionally low sell tax amplified by the absence of a buy tax, improves attractiveness for holders and bolsters market liquidity. Its widespread adoption is demonstrated by a substantial holder count. However, the LP holder count is less than desirable, suggesting concentration of liquidity ownership. Notably, IMX being mintable could pose dilution risks, but its presence on a Dex and its open-source attribute enhance transparency, making the token more trustworthy. High total supply may potentially dampen the asset's value over time. Conclusively, IMX shows potential with certain risk elements demanding careful evaluation before investment.

Conclusion Overview

Overview	Notes	Result
Honeypot	The contract owner can drain the funds from contract	✓ False
Anti whale check	Features preventing whales to manipulate the Token	✗ False
Opensource	The code of the contract is public	✓ True
Ownership renounced	Contract owner has renounced ownership	✗ False
Buy tax	Fees incurred when buying the token	✓
Sell tax	Fees incurred when selling the token	✓
High Severity Issues	Number of High severity issues	1
Medium Severity Issues	Number of Medium severity issues	1
Mintable	Can mint new tokens	✗ True
Blacklist	Owner can blacklist users	✓ False
Holders	Total wallets holding the token	✓ 64892
LP holder	Total wallets holding the token	✓ 5

Glossary

1. Honey pot:

A cybersecurity strategy involving the deployment of decoy systems or resources that appear vulnerable to attackers. The goal is to attract and monitor malicious activity, gaining insights into hacker tactics and motives for enhanced security.

2. Blacklist:

catalog of known malicious entities, such as IP addresses, domain names, or applications, used to deny access or privileges. Blacklists safeguard systems and networks by blocking or restricting interaction with these entities, preventing potential threats or unauthorized access.

3. Ownership privileges:

Ownership is the legal right to possess, use, and dispose of property or assets, typically accompanied by control, responsibility, and the ability to transfer or sell.

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:

What is LLM in simple words? 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.