Assure DeFi®

THE VERIFICATION GOLD STANDARD

VERIFIED BY ASSURE DEFI ® INTEGRITY ✦ TRUST ✦ CREDIBILITY

# Security Assessment

# LegalX

Date: 01/03/2025

Audit Status: FAIL

Audit Edition: Standard+

ASSURE DEFI®

THE VERIFICATION GOLD STANDARD

# Risk Analysis

## Vulnerability summary

| Classification | Description |
|---|---|
| 🔴 High | High-level vulnerabilities can result in the loss of assets or manipulation of data. |
| 🟠 Medium | Medium-level vulnerabilities can be challenging to exploit, but they still have a considerable impact on smart contract execution, such as allowing public access to critical functions. |
| 🟡 Low | Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored. |
| 🟢 Informational | Informational vulnerabilities, code style violations, and informational statements do not affect smart contract execution and can typically be disregarded. |

## Executive Summary

According to the Assure assessment, the Customer's smart contract is **Poorly Secured.**

| | | | |
|---|---|---|---|
| Insecure | **Poorly Secured** | Secured | Well Secured |

# Scope

## Target Code And Revision

For this audit, we performed research, investigation, and review of the LegalX contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

## Target Code And Revision

| | |
|---|---|
| **Project** | Assure |
| **Language** | Solidity |
| **Codebase** | LegalXFlat.sol [SHA256]:<br>f85461723b0a32ab025493c2694e8246caa6219c87989a59b3f1cfdfe39ecd2c |
| **Audit Methodology** | Static, Manual |

# Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

| Category | Item |
|---|---|
| Code review & Functional Review | ● Compiler warnings.<br>● Race conditions and Reentrancy. Cross-function race conditions.<br>● Possible delays in data delivery.<br>● Oracle calls.<br>● Front running.<br>● Timestamp dependence.<br>● Integer Overflow and Underflow.<br>● DoS with Revert.<br>● DoS with block gas limit.<br>● Methods execution permissions.<br>● Economy model.<br>● Private user data leaks.<br>● Malicious Event log.<br>● Scoping and Declarations.<br>● Uninitialized storage pointers.<br>● Arithmetic accuracy.<br>● Design Logic.<br>● Cross-function race conditions.<br>● Safe Zeppelin module.<br>● Fallback function security.<br>● Overpowered functions / Owner privileges |

# AUDIT OVERVIEW

## ◆ ◆ ◆ HIGH

### 1. Claim Wait Update Logic Bug

**Function**: updateClaimWait (DividendTracker)

**Issue**: The condition newClaimWait != claimWait prevents any update because the new value must equal the current value, making it impossible to change the claim wait time.

**Recommendation**: Remove or adjust the equality check so that the only constraints are the lower and upper bounds (e.g., between 1 minute and 1 day).

## ◆ ◆ ◆ MEDIUM

### 1. Dividend Balance Update Complexity

**Function**: _update, mintBalance, burnBalance (DividendPayingToken & DividendTracker)

**Issue**: The use of an unconditional revert in _update combined with an override in DividendTracker creates a complex inheritance pattern that may trigger unexpected reverts if not used properly.

**Recommendation**: Clearly document the inheritance chain and ensure that internal balance updates use the explicit super._update call. Consider refactoring to a clearer pattern.

### 2. Dividend Processing Gas Limitation

**Function**: process (DividendTracker)

**Issue**: Iterating over the token holders using an iterable mapping might run out of gas when the number of holders is large, potentially leading to DoS-like behavior.

**Recommendation**: Consider batching or chunking the dividend processing loop and optimize the iteration to handle large numbers of token holders.

### 3. Non-standard Swapping Guard

**Function**: _update, swapAndProcessTokens, processDividendTracker (LegalXToken)

**Issue**: The bit–shift based swapping guard is non–standard and opaque, which can lead to reentrancy vulnerabilities if mismanaged.

**Recommendation**: Replace the bit–shift mechanism with a standard reentrancy guard (e.g., OpenZeppelin's ReentrancyGuard) for clearer and more robust protection.

### 4. Potential Reentrancy in External Calls

**Function**: swapAndProcessTokens, swapTokensForETH (LegalXToken)

**Issue**: External calls—particularly the low-level ETH transfer to the marketing wallet—could be exploited by a malicious recipient if the reentrancy guard is bypassed.

**Recommendation**: Introduce a reentrancy guard on the fee-processing functions and consider using a pull–payment pattern for ETH transfers. Verify that external call targets are trusted.

## LOW

### 1. Use of tx.origin in Logging

**Function**: processDividendTracker (LegalXToken)

**Issue**: Using tx.origin for logging purposes is discouraged since it can be spoofed in complex call chains, even though it is only used for informational purposes.

**Recommendation**: Replace tx.origin with msg.sender in event emissions for improved clarity and security.

## INFORMATIONAL

### 1. The Solidity version pragma syntax is incorrect and should be corrected.

**Issue**: The top of the code contains a pragma line.

pragma solidity =0.8.24 ^0.8.0 ^0.8.20;

This syntax is not standard (or even valid) because it mixes different version specifiers.

**Recommendation**: Use a single version specifier.

# Annexes

The configuration (5% total fee split as 3% for the legal/marketing wallet and 2% for reflections) is implemented via the state variables (rewardTokenFee = 2, marketingFee = 3, totalFees = 5) and matches LegalX team specification.

# Technical Findings Summary

## Findings

| Vulnerability Level | Total | Pending | Not Apply | Acknowledged | Partially Fixed | Fixed |
|---|---|---|---|---|---|---|
| 🔴 High | 1 | | | | | |
| 🟠 Medium | 4 | | | | | |
| 🟡 Low | 1 | | | | | |
| 🟢 Informational | 1 | | | | | |

# Assessment Results

## Score Results

| Review | Score |
|---|---|
| **Global Score** | **70/100** |
| Assure KYC | Not completed |
| Audit Score | 70/100 |

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project must pass and provide all the data needed for the assessment. Our Passing Score has been changed to 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below. The Global Score is a combination of the evaluations obtained between having or not having KYC and the type of contract audited together with its manual audit.

## Audit Failed

Following our comprehensive security audit of the token contract for the LegalX project, we inform you that the cybersecurity audit has failed due to multiple critical issues identified during the review, which pose significant risks to the contract's functionality and security. Immediate remediation is required to address these vulnerabilities.

# Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocating for the Project, and users relying on this report should not consider this as having any merit for financial adLegalX in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment adLegalX, nor should it be used to signal that any person reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audit completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

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