



Zest Token Security Review

Conducted by:
Clarity Alliance

April 2nd, 2026



Contents

1. About Clarity Alliance	2
2. Disclaimer	3
3. Introduction	3
4. About Zest Token	4
5. Risk Classification	4
5.1. Impact	4
5.2. Likelihood	5
5.3. Action required for severity levels	5
6. Security Assessment Summary	6
7. Executive Summary	8
8. Findings	9

1. About Clarity Alliance

Clarity Alliance is a team of expert whitehat hackers specialising in securing protocols on Stacks.

They have disclosed vulnerabilities that have saved millions in live TVL and conducted thorough reviews for some of the largest projects across the Stacks ecosystem.

Learn more about Clarity Alliance at clarityalliance.org

2. Disclaimer

This report is not, and should not be considered, an endorsement or disapproval of any project, team, product, or asset, nor does it reflect on their economics, value, business model, or legal compliance. It does not provide any warranty regarding the absolute bug-free nature or functionality of the analyzed technology.

Nothing in this report should be used to make investment or participation decisions. It does not constitute investment advice. Instead, it reflects an extensive assessment process intended to help clients improve code quality and reduce the inherent risks associated with cryptographic tokens and blockchain systems.

Blockchain technology presents ongoing and significant risk, and each company or individual remains responsible for their own due diligence and security posture. Clarity Alliance aims to reduce attack vectors and technological uncertainty but does not guarantee the security or performance of any system we review.

All assessment services are subject to dependencies and active development. Your access to and use of any services, reports, or materials is at your sole risk on an as-is and as-available basis.

Cryptographic tokens are emergent technologies with high technical uncertainty, and assessment results may include false positives, false negatives, or other unpredictable outcomes. Smart contracts may depend on multiple external parties, remain vulnerable to internal or external exploitation, and may carry elevated risks if owner privileges remain active. Accordingly, Clarity Alliance does not guarantee the explicit security of any audited smart contract, regardless of the reported verdict.

3. Introduction

A time-boxed security review of Zest Token, where Clarity Alliance reviewed the scope and provided insights on improving the protocol.

4. About Zest Token

The upcoming **Zest Token** is implemented as a fully immutable contract that mints a fixed supply at deployment via a single-use constructor, with no owner, admin privileges, or upgrade path, relying entirely on unmodified OpenZeppelin v5 standards for all core functionality.

5. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

5.1. Impact

- High - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- Medium - only a small amount of funds can be lost (such as leakage of value) or a core functionality of the protocol is affected.
- Low - can lead to any kind of unexpected behavior with some of the protocol's functionalities that's not so critical.

5.2. Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.
- Medium - only a conditionally incentivized attack vector, but still relatively likely.
- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

5.3. Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

6. Security Assessment Summary

Contract Under Review

- | **Contract Name** | ZestToken |
- | **Token Name** | Zest |
- | **Token Symbol** | ZEST |
- | **Contract Address** | `0x5506599c722389A60580B5213ea1Da60D64754a1` |
- | **Chain** | BSC Mainnet (Chain ID 56) |
- | **Source Code** | BscScan (verified) |
- | **Solidity Version** | 0.8.24 |
- | **Optimizer** | Enabled (200 runs) |
- | **Compiler Standard** | OpenZeppelin Contracts v5 |
- | **Deployment Tx** |
`0xa8b65fdf675a6bffbdc9c2ac869a98dd11cb319aa4f42a570dbfc932498ab0d` |
- | **Deployment Block** | 82083470 |

Source Files

File	Description	LoC
<code>src/ZestToken.sol</code>	Token contract - ERC20 + ERC20Permit inheritance, constructor only	11

Dependencies

Dependency	Version / Commit
OpenZeppelin Contracts	v5 @ <code>fcbae5394ae8ad52d8e580a3477db99814b9d565</code>

Token Parameters

Parameter	Value
Name	Zest
Symbol	ZEST
Decimals	18
Total Supply	1,000,000,000 ZEST (fixed)
Supply Mechanism	Minted once in constructor to deployer

Scope Details

- `src/ZestToken.sol` - constructor behavior, inheritance correctness, deployed bytecode match against source at the specified commit
- Confirmation that no custom overrides modify inherited OpenZeppelin behavior
- Confirmation of immutability: no owner, no admin, no upgrade path

Threat Model

The contract is immutable and has no privileged functions. The only custom logic is the constructor, which mints a fixed supply once and cannot be invoked again. All other behavior is delegated to audited OpenZeppelin v5 implementations. The attack surface is limited to: (1) the constructor correctly initializing state, and (2) the inherited OpenZeppelin code operating as documented.

7. Executive Summary

Over the course of the security review, Clarity Alliance engaged with Zest to review Zest Token. In this period of time a total of **0** issues were uncovered.

Protocol Summary

Protocol Name	Zest
Date	April 2nd, 2026

Findings Count

Severity	Amount
Total Findings	0

8. Findings

The ZestToken contract is a minimal implementation that inherits exclusively from OpenZeppelin Contracts v5, the industry standard Solidity library for ERC20 tokens. The contract contains no custom business logic beyond a constructor that assigns the token name, assigns the token symbol, and mints a fixed supply of 1,000,000,000 tokens to the deployer. All inherited behavior, including transfers, approvals, allowances, and EIP2612 permit functionality, is delegated to OpenZeppelin's audited and formally reviewed implementations without modification or override.

The contract is fully immutable and authority free by design. There is no owner, no admin role, no access control mechanism, and no upgradeability pattern such as a proxy or beacon. No post deployment minting, burning, pausing, blacklisting, or fee adjustment functions exist, and none can be introduced. Following deployment, the entire token supply was transferred from the deployer to a Gnosis Safe multisig, and the deployer retains zero privileged capabilities over the contract or its supply. This design eliminates the most common classes of token contract risk, including owner key compromise, malicious upgrades, arbitrary minting, and transfer pausing, by structurally removing the functions themselves rather than relying on operational controls.

Given the contract's minimal surface area, its exclusive reliance on audited OpenZeppelin v5 standards, and the absence of any privileged roles or custom logic, we consider the deployed contract to be free of material risk beyond that of the underlying OpenZeppelin library, which is among the most widely reviewed and production tested codebases in the Ethereum ecosystem.