Audit Master

Professional Audit Reports

Audit Report

Security Audit Report TokenFactory_V1

Project Name PolyMint

Smart Contract Address 0xcBEF00938F786Ab9B8Bb617Af767E3DD78B3CF39

Security Audit Provider Audit Master

Audit Date 2026-01-12

1. Scope of Audit

The audit covers the following smart contracts

TokenFactory_V1 Factory contract to create and verify custom ERC20 tokens.

CustomToken Minimal ERC20 implementation with owner-controlled supply and transfer functions.

The audit focused on security, correctness, and best practices in Solidity development.

2. Key Findings

1 Ownership and Access Control

onlyOwner modifier is applied correctly on sensitive functions setCreationFee, setVerifyFee, setMaxSupply, transferFactoryOwnership, transferOwnership.

No public functions allow unauthorized administrative changes.

Suggestion Consider renouncing ownership or using a multi-signature wallet for added security in production.

2 Fee Management

Creation and verification fees are handled properly.

Ether payments are transferred to the factory owner using payableowner.transfermsg.value.

Safe for small amounts, but consider pull-over-push pattern for large-scale usage to avoid reentrancy risks.

3 Token Creation

createToken enforces _totalSupply maxSupply.

Tokens are instantiated via new CustomToken... and tracked in allTokens array.

Emits TokenCreated event correctly for transparency.

4 Token Verification

verifyToken ensures a token is verified only once.

Emits TokenVerified event correctly.

5 CustomToken Implementation

ERC20 functions transfer, approve, transferFrom are implemented manually.

Safe arithmetic in Solidity 0.8.19 built-in overflow checks.

balanceOf and allowance are tracked correctly.

Ownership transfer is handled securely.

Suggestion Consider implementing increaseAllowance decreaseAllowance to prevent approval race conditions.

6 Gas Efficiency

Factory tracks all tokens in an array allTokens.

Can become expensive in storage if a large number of tokens are created.

Consider pagination or mapping-based enumeration for scalability.

7 Security Risks

Risk	Severity	Notes
Reentrancy on Ether transfers	Low	Using .transfer is safe, but large payments could fail if gas exceeds 2300.
Centralized Ownership	Medium	Single owner controls fees, max supply, and verification. Multi-sig recommended.
ERC20 Approval Race	Low	Manual approve may allow double-spend in some edge cases.

7. Summary Score

The contracts are well-written and secure for standard use. Minor improvements suggested for scalability and multi-signature safety.

Overall Security Score 92100

Strengths

Correct access control on critical functions.

Proper event logging for transparency.

Safe ERC20 implementation using Solidity 0.8 features.

Recommendations

Consider a multi-signature for owner functions.

Implement increaseAllowance decreaseAllowance in CustomToken.

Use pull-payment pattern for handling large Ether transfers.

Review gas cost if allTokens grows significantly.