Soneium ShibuyaToken Audit Report

Prepared by Cyfrin

Version 2.0

Lead Auditors

Hans

December 23, 2024

Contents

1 About Cyfrin

Cyfrin is a Web3 security company dedicated to bringing industry-leading protection and education to our partners and their projects. Our goal is to create a safe, reliable, and transparent environment for everyone in Web3 and DeFi. Learn more about us at cyfrin.io.

2 Disclaimer

The Cyfrin team makes every effort to find as many vulnerabilities in the code as possible in the given time but holds no responsibility for the findings in this document. A security audit by the team does not endorse the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the solidity implementation of the contracts.

3 Risk Classification

	Impact: High	**Impact: Medium**	**Impact: Low**
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

4 Protocol Summary

ShibuyaToken contract is a key component in bridging ASTR (Astar Network's native token) from Astar L1 to Soneium L2. Soneium is a Layer 2 solution built on OP Stack technology, developed through a collaboration between Sony Group Corporation and Startale.

4.1 Architecture Overview

The bridging system consists of two main components:

1. ASTR L1 to Soneium L2 Bridge:
   - Utilizes Chainlink's CCIP for secure cross-chain transfers
   - Implements wrap-native functionality for improved UX
   - Single approval transaction design
2. ShibuyaToken Contract (Soneium L2):
   - Supports CCIP integration for L1-L2 bridging by adhering to the CCIP's Cross-Chain Token Requirements
   - Enables further bridging to other Superchain L2s through SuperchainERC20 standard (ERC7802) compliance
   - Features upgradeable proxy pattern
   - Includes role-based access control

4.2 ShibuyaToken

ShibuyaToken is an upgradeable ERC20 token contract with minting, burning, and role-based access control capabilities, specifically designed to support cross-chain operations.

4.2.1 ACTORS & ROLES

1. Owner

- Primary administrator of the contract
- Can grant/revoke roles
- Can initiate ownership transfer
- Can authorize contract upgrades
- Set during initialization

2. Pending Owner

- Address proposed to become the new owner
- Must accept ownership via acceptOwnership()

3. Role Holders

- MINTER_ROLE: Can mint new tokens and perform crosschain mints
- BURNER_ROLE: Can burn tokens and perform crosschain burns

4.2.2 PUBLIC FUNCTIONS

Initialization:

```
initializeV3(address defaultAdmin)
```

- Initializes contract with "Shibuya Token" (SBY)
- Sets initial owner
- Revokes `DEFAULT_ADMIN_ROLE` from the previous admin

Token Operations:

1. Minting:

```
mint(address account, uint256 amount)
crosschainMint(address _to, uint256 _amount)
```

2. Burning:

```
burn(uint256 amount)
burn(address account, uint256 amount)
burnFrom(address account, uint256 amount)
crosschainBurn(address _from, uint256 _amount)
```

Role Management:

```
grantMintAndBurnRoles(address minAndBurner)
grantMintRole(address minter)
grantBurnRole(address burner)
revokeMintAndBurnRoles(address minAndBurner)
revokeMintRole(address minter)
revokeBurnRole(address burner)
```

```
grantRole(bytes32 role, address account)
revokeRole(bytes32 role, address account)
```

Ownership Management:

```
transferOwnership(address to)
acceptOwnership()
owner()
```

4.2.3 SECURITY POINTS

1. Owner privileges are extensive and centralized
2. No pause mechanism for emergencies
3. No explicit transfer restrictions
4. Multiple burn paths need careful access control

5 Audit Scope

This security review analyzes the `ShibuyaToken` contract implemented in `contracts/Shibuya.sol`.

6 Executive Summary

Over the course of 5 days, the Cyfrin team conducted an audit on the Soneium ShibuyaToken smart contracts provided by Startale. In this period, a total of 6 issues were found.

This security review evaluates the `ShibuyaToken` contract with a primary focus on its Soneium side token contract security, correctness of Superchain ERC20 (ERC7802) implementation. Additional attention was given to CCIP integration and overall architecture as secondary focus areas.

The token contract demonstrates solid implementation practices with comprehensive test coverage. No critical or high-severity issues were identified during the assessment across any of the focus areas. The findings are primarily LOW and INFO severity issues that, while not presenting immediate security risks, could enhance the contract's robustness if addressed:

Notable Observations:

- Minor validation gaps in access control management
- Some opportunities for code optimization and documentation improvements
- Non-critical input validation enhancements for cross-chain operations

The contract's test suite is comprehensive with high coverage, indicating strong quality assurance practices. The Superchain ERC20 implementation correctly follows the standard specifications, and the bridging mechanism through CCIP is properly integrated. While the suggested improvements would further enhance the contract's quality, they do not impact the core functionality or security of the implementation.

Summary

Project Name	Soneium ShibuyaToken
Repository	ccip-contracts-registration
Commit	b8491f385c80. . .
Fix Commit	129bc6a11eff. . .
Audit Timeline	Dec 12th - Dec 18th
Methods	Manual Review, Fuzzing

Issues Found

Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	1
Informational	5
Gas Optimizations	0
Total Issues	6

Summary of Findings

[L-1] Lack of validation for default admin revocation during initialization	Resolved
[I-1] Incorrect documentation for mint function	Resolved
[I-2] Redundant owner-based access control implementation	Acknowledged
[I-3] Missing zero amount validation in crosschain operations	Resolved
[I-4] Incomplete ERC20 interface support	Resolved
[I-5] Parameter naming inconsistency	Resolved

7 Findings

7.1 Low Risk

7.1.1 Lack of validation for default admin revocation during initialization

Description: In `ShibuyaToken::initializeV3()`, the contract attempts to revoke `DEFAULT_ADMIN_ROLE` from the provided `defaultAdmin` parameter. Based on the deployed contract on testnet and team communication, the intention was to remove the role from the original default admin address during the upgrade process.

However, the current implementation has two issues:

1. It revokes the role without first verifying if the address actually possesses the role in the previous version
2. The accompanying comments are misleading, stating "this is to make sure that the default admin is the owner and remove the default admin role from the deployer". This implies the deployer always has the default admin role, which isn't necessarily true.

```
Shibuya.sol
36:         // this is to make sure that the default admin is the owner and remove the default admin
↪  role from the deployer
37:         // so this means that the deployer will not have any role in the contract
38:         // the the DEFAULT_ADMIN_ROLE will not have any role in the contract
39:         // and the owner will be performing the role of the admin
40:         _revokeRole(DEFAULT_ADMIN_ROLE, defaultAdmin);
```

Additionally, there's a typographical error in the comments at line 38 where "the" is repeated.

Recommended Mitigation:

- Add an explicit check if `defaultAdmin` has `DEFAULT_ADMIN_ROLE` before revoking
- Document clearly in the upgrade guide about role requirements
- Fix minor mistakes in comments

Startale: Fixed in commit 01789b.

Cyfrin: Verified.

7.2 Informational

7.2.1 Incorrect documentation for mint function

Description: The `mint` function's documentation incorrectly states it "disallows burning from address(0)" when it should be "disallows minting to address(0)".

Recommended Mitigation: Update the documentation to correctly reflect the function's behavior

Startale: Fixed in PR 8.

Cyfrin: Verified.

7.2.2 Redundant owner-based access control implementation

Description: The contract implements a custom ownership system alongside `AccessControlUpgradeable`, but fails to utilize the latter's built-in access control mechanisms effectively. Since the owner is not granted the `DEFAULT_ADMIN_ROLE`, the contract is forced to override several public functions from `AccessControlUpgradeable` to maintain proper access control.

This design choice implies potential issues:

1. Unnecessarily duplicates access control functionality that already exists in `AccessControlUpgradeable`
2. Requires overriding `grantRole()` and `revokeRole()` functions
3. Inconsistently handles role management by missing the override for `renounceRole()`
4. Increases code complexity and potential for access control confusion

While the current implementation is functional, it introduces unnecessary complexity and potential maintenance challenges.

Recommended Mitigation: Instead of implementing a separate ownership system we recommend the team to consider:

1. Grant the `DEFAULT_ADMIN_ROLE` to the owner during initialization
2. Remove custom ownership implementation
3. Utilize `AccessControlUpgradeable`'s built-in role management functions
4. Remove unnecessary function overrides

Startale: Acknowledged.

Cyfrin: Acknowledged.

7.2.3 Missing zero amount validation in crosschain operations

Description: The `crosschainMint()` and `crosschainBurn()` functions don't validate for zero amounts, which could lead to unnecessary event emissions.

Recommended Mitigation: Add zero amount checks and revert if amount is zero.

Startale: Fixed in commit fa77e9.

Cyfrin: Verified.

7.2.4 Incomplete ERC20 interface support

Description: The `supportsInterface()` function doesn't declare support for `IERC20` interface ID despite implementing ERC20 functionality. This could cause issues with interface detection in some integration scenarios.

Recommended Mitigation: Add support for `type(IERC20).interfaceId` in the `supportsInterface` function.

Startale: Fixed in commit cb5b05.

Cyfrin: Verified.

7.2.5 Parameter naming inconsistency

Description: The parameter name ’minAndBurner’ in the `grantMintAndBurnRoles()` function has a typo.

```
function grantMintAndBurnRoles(address minAndBurner)
```

Recommended Mitigation: Correct the parameter name to ’mintAndBurner’.

Startale: Fixed in commit 669197.

Cyfrin: Verified.