# Smart Contract Security Audit Report

---

## Socket

# 1. Contents

# 2. General Information

This report contains information about the results of the security audit of the Socket.Tech (hereafter referred to as "Customer") smart contracts, conducted by Decurity in the period from 03/25/2024 to 03/27/2024.

## 2.1. Introduction

Tasks solved during the work are:

- Review the protocol design and the usage of 3rd party dependencies,
- Audit the contracts implementation,
- Develop the recommendations and suggestions to improve the security of the contracts.

## 2.2. Scope of Work

The audit scope included the contracts in the following repository: https://github.com/SocketDotTech/socket-ll-contracts/blob/feat/scroll-native-bridge. Initial review was done for the commit 7ee4cfc27d258f5c92283e2546bc9bd7e39d6ee6. The remediation review was done for the commit 102128a6f761f3f5d3edc7e621ea26667e141475.

The following contracts have been tested:

- ScrollImpl

## 2.3. Threat Model

The assessment presumes the actions of an intruder who might have the capabilities of any role (an external user, token owner, token service owner, or a contract).

The main possible threat actors are:

- User,
- Protocol owner,
- Relayer,

- Token owner/contract.

## 2.4. Weakness Scoring

An expert evaluation scores the findings in this report, and the impact of each vulnerability is calculated based on its ease of exploitation (based on the industry practice and our experience) and severity (for the considered threats).

## 2.5. Disclaimer

Due to the intrinsic nature of the software and vulnerabilities and the changing threat landscape, it cannot be generally guaranteed that a certain security property of a program holds.

Therefore, this report is provided “as is” and is not a guarantee that the analyzed system does not contain any other security weaknesses or vulnerabilities. Furthermore, this report is not an endorsement of the Customer’s project, nor is it an investment advice.

That being said, Decurity exercises the best effort to perform its contractual obligations and follow the industry methodologies to discover as many weaknesses as possible and maximize the audit coverage using limited resources.

# 3. Summary

As a result of this work, we have discovered a single issue with direct impact, which has been fixed and re-tested in the course of the work.

The other suggestions included fixing the low-risk issues and some best practices (see Security Process Improvement).

## 3.1. Suggestions

The table below contains the discovered issues, their risk level, and their status as of 01 April, 2024.

*Table. Discovered weaknesses*

| Issue | Contract | Risk Level | Status |
|---|---|---|---|
| Usage of the deprecated SELFDESTRUCT | src/bridges/BridgeImplBase.sol | Low | Fixed |
| Event not emitted for ERC20 swapAndBridge | src/bridges/scroll/ScrollBridgeImpl.sol | Low | Fixed |
| Insufficient tests | test/solidity/bridges/scroll/ScrollERC20.t.sol, test/solidity/bridges/scroll/ScrollEth.t.sol | Info | Acknowledged |
| Unnecessary public variable | src/scroll/ScrollBridgeImpl.sol | Info | Fixed |
| Redundant emit statements | src/bridges/scroll/ScrollBridgeImpl.sol | Info | Fixed |

# 4. General Recommendations

This section contains general recommendations on how to improve the overall security level.

The Findings section contains technical recommendations for each discovered issue.

## 4.1. Security Process Improvement

The following is a brief long-term action plan to mitigate further weaknesses and bring the product security to a higher level:

- Keep the whitepaper and documentation updated to make it consistent with the implementation and the intended use cases of the system,
- Perform regular audits for all the new contracts and updates,
- Ensure the secure off-chain storage and processing of the credentials (e.g. the privileged private keys),
- Launch a public bug bounty campaign for the contracts.

# 5. Findings

## 5.1. Usage of the deprecated SELFDESTRUCT

**Risk Level**: Low

**Status**: Fixed in the commit 102128a6.

**Contracts**:

- src/bridges/BridgeImplBase.sol

**Location**: Lines: 115.

**Description:**

The `killme` function calls the `selfdestruct` function which was deprecated by EIP-6780.

**Remediation:**

To implement the contract pause or destruction, use other methods such as `Pausable` pattern.

**References:**

- https://eips.ethereum.org/EIPS/eip-6780

## 5.2. Event not emitted for ERC20 swapAndBridge

**Risk Level**: Low

**Status**: Fixed in the commit 27a595e8.

**Contracts**:

- src/bridges/scroll/ScrollBridgeImpl.sol

**Location**: Lines: 293.

**Description:**

The `swapAndBridge` function does not emit `SocketBridge` event when the target token is not native.

As a result, the analytical tools may report wrong information. Note that these events do not affect bridging because the funds are sent immediately to the Scroll native bridge, and their relayers don't parse the events issues in the Socket's contracts.

**Remediation:**

Remove the `return` statement from the line 293 and replace `NATIVE_TOKEN_ADDRESS` with `token` in the `emit` statement on the line 298.

## 5.3. Insufficient tests

**Risk Level**: Info

**Status**: Acknowledged

**Contracts**:

- test/solidity/bridges/scroll/ScrollERC20.t.sol,
- test/solidity/bridges/scroll/ScrollEth.t.sol

**Description:**

The tests don't contain the assertions for the emitted events. This may lead to integration bugs if no additional tests involving relayers are done.

**Remediation:**

Ensure full test coverage and add assertions for the emitted events.

## 5.4. Unnecessary public variable

**Risk Level**: Info

**Status**: Fixed in the commit bbbd4767.

**Contracts**:

- src/scroll/ScrollBridgeImpl.sol

**Location**: Lines: 13.

**Description:**

There is an unnecessary public getter for the `UINT_MAX` variable which causes additional bytecode creation and gas costs.

**Remediation:**

Replace the public constant with private immutable.

## 5.5. Redundant emit statements

**Risk Level**: Info

**Status**: Fixed in the commit bbbd4767.

**Contracts**:

- src/bridges/scroll/ScrollBridgeImpl.sol

**Location**: Lines: 181-189.

**Description:**

There're 2 `emit` statements in the `bridgeAfterSwap` function that are identical except the token address. In case when the token address is equal `NATIVE_TOKEN_ADDRESS`, a separate emit statement is used.

This does not make sense because the `token` variable already contains the same value.

**Remediation:**

Replace the 2 statements with a single event emit after the deposit (i.e. outside the `if-else` statement).

# 6. Appendix

## 6.1. About us

The Decurity team consists of experienced hackers who have been doing application security assessments and penetration testing for over a decade.

During the recent years, we've gained expertise in the blockchain field and have conducted numerous audits for both centralized and decentralized projects: exchanges, protocols, and blockchain nodes.

Our efforts have helped to protect hundreds of millions of dollars and make web3 a safer place.