Spool

SMART CONTRACT AUDIT

ZOKYO.

December 9th, 2021 | v. 1.0

PASS

Zokyo's Security Team has concluded that this smart contract passes security qualifications to be listed on digital asset exchanges.

TECHNICAL SUMMARY

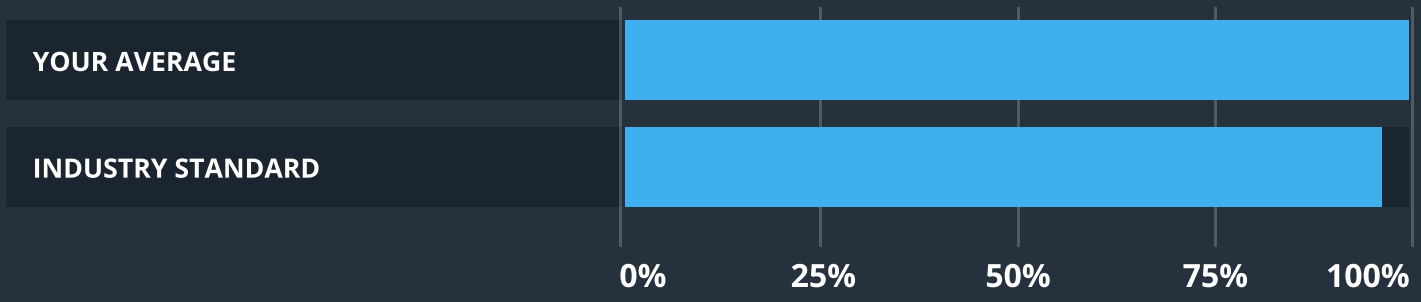
This document outlines the overall security of the SpoolFi DAO Token smart contracts, evaluated by Zokyo's Blockchain Security team.

The scope of this audit was to analyze and document the SpoolFi smart contract codebase for quality, security, and correctness.

Contract Status

There were no critical and high issues found during the audit.

Testable Code



The testable code is 100%, which is above the industry standard of 95%.

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a security of the contract we at Zokyo recommend that the SpoolFi DAO put in place a bug bounty program to encourage further and active analysis of the smart contract.

TABLE OF CONTENTS

AUDITING STRATEGY AND TECHNIQUES APPLIED

The SpoolFi smart contract's source code was taken from the repositories provided by the SpoolFi DAO: https://github.com/SpoolFi/spool-dao-token

Initial commits (audited): 3ba423580876c1965d845fa64c88911f34c0a74d

Post-audit commits (verified): 473149182d80c86eda6e46e20582fb2a4d5c8578

Within the scope of this audit Zokyo auditors have reviewed the following contract(s):

- SpoolDaoToken.sol

Throughout the review process, care was taken to ensure that the contract:

- Implements and adheres to existing standards appropriately and effectively;
- Documentation and code comments match logic and behavior;
- Follows best practices in efficient use of resources, without unnecessary waste;
- Uses methods safe from reentrance attacks;
- Is not affected by the latest vulnerabilities;
- Whether the code meets best practices in code readability, etc.

Zokyo's Security Team has followed best practices and industry-standard techniques to verify the implementation of SpoolFi smart contracts. To do so, the code is reviewed line-by-line by our smart contract developers, documenting any issues as they are discovered. Part of this work includes writing a unit test suite. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

1	Due diligence in assessing the overall code quality of the codebase.	3	Testing contract logic against common and uncommon attack vectors.
2	Cross-comparison with other, similar smart contracts by industry leaders.	4	Thorough, manual review of the codebase, line-by-line.

EXECUTIVE SUMMARY

There were no critical issues found during the audit. The code overall follows coding standards and has high quality. The only issues found refer to the slight optimisations and increasing of the code readability.

Nevertheless, all issues were successfully fixed by SpoolFi DAO.

STRUCTURE AND ORGANIZATION OF DOCUMENT

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged “Resolved” or “Unresolved” depending on whether they have been fixed or addressed. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

Critical

The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

High

The issue affects the ability of the contract to compile or operate in a significant way.

Medium

The issue affects the ability of the contract to operate in a way that doesn’t significantly hinder its behavior.

Low

The issue has minimal impact on the contract’s ability to operate.

Informational

The issue has no impact on the contract’s ability to operate.

COMPLETE ANALYSIS

INFORMATIONAL | RESOLVED

Unnecessary multiplication.
Lines 26, minting function utilizes multiplication by 1 ether, though direct literal usage is allowed. Removing extra operation saves a bit of gas.

Recommendation:
Multiplication can be removed

INFORMATIONAL | RESOLVED

Utilize constant
Lines 26, number of tokens minted may be moved to the public constant instead of constructing a variable - such a move will save a bit of gas during deployment.

Recommendation:
Consider usage of the public constant.

INFORMATIONAL | RESOLVED

Shadowed variable
In constructor, local variable “owner” shadows the call for the owner() function of the Ownable contract.
https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Recommendation:
Consider renaming (for example to _owner).

	SpoolDaoToken
Re-entrancy	Pass
Access Management Hierarchy	Pass
Arithmetic Over/Under Flows	Pass
Delegatecall Unexpected Ether	Pass
Default Public Visibility	Pass
Hidden Malicious Code	Pass
Entropy Illusion (Lack of Randomness)	Pass
External Contract Referencing	Pass
Short Address/ Parameter Attack	Pass
Unchecked CALL Return Values	Pass
Race Conditions / Front Running	Pass
General Denial Of Service (DOS)	Pass
Uninitialized Storage Pointers	Pass
Floating Points and Precision	Pass
Tx.Origin Authentication	Pass
Signatures Replay	Pass
Pool Asset Security	Pass

CODE COVERAGE AND TEST RESULTS FOR ALL FILES

Tests written by Zokyo Secured team

As part of our work assisting SpoolFi DAO in verifying the correctness of their contract code, our team was responsible for writing integration tests using Hardhat testing framework.

Tests were based on the functionality of the code, as well as review of the SpoolFi contract requirements for details about issuance amounts and how the system handles these.

SpoolDaoToken

- ✓ Token has correct name and symbol
- ✓ Token has correct supply and correct supply owner
- ✓ Token can be transferred
- ✓ Token can be paused and unpaused by owner
- ✓ Token cannot be transferred during the pause
- ✓ Ownership can transferred by the owner

6 passing (1s)

FILE	% STMTS	% BRANCH	% FUNCS	% LINES	UNCOVERED LINES
contracts/	100	100	100.00	100	
SpoolDaoToken.sol	100	100	100.00	100	
All files	**100**	**100**	**100.00**	**100**	

We are grateful to have been given the opportunity to work with the SpoolFi.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them.

Zokyo's Security Team recommends that the SpoolFi put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

ZOKYO.