

|  |
| --- |
| AUDIT REPORT  2022 |

|  |  |  |
| --- | --- | --- |
| SWAN TOKEN SECURITY ADUIT |  |  |
|  |  |  |

TABLE OF CONTENTS

[OVERVIEW 3](#_Toc104092425)

[Project Summary 3](#_Toc104092426)

[Audit Summary 3](#_Toc104092427)

[Vulnerability Summary 4](#_Toc104092428)

[Audit Scope 5](#_Toc104092429)

[FINDINGS 6](#_Toc104092431)

[Project Description 6](#_Toc104092432)

[Project Structure 6](#_Toc104092433)

[Automated Test using Cargo 6](#_Toc104092434)

[Manual Test Comparing to SPL 7](#_Toc104092435)

[Others 8](#_Toc104092435)

[CONCLUSION 9](#_Toc104092436)

[Overview 9](#_Toc104092437)

[Problems and Recommendation 9](#_Toc104092438)

[DISCLAIMER 10](#_Toc104092443)

# Summary

This report has been prepared for Custom Solana Program Library named SWAN to discover issues and vulnerabilities in the source code of Swan Token forked from spl that were not part an officially recognized library. A comprehensive examination has been performed, utilizing Tooling Analysis and Manual Review techniques by smart contract security expert.

The auditing process pays special attention to the following considerations:

* Testing the smart contracts against both common and uncommon attack vectors.
* Assessing the codebase to ensure compliance with current best practices and industry standards.
* Ensuring contract logic meets the general specifications and intentions of the standard design patterns
* Cross referencing contract structure and implementation against similar contract produced by industry leaders.
* Through line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspectives:

* Enhancing general coding practices for better structures of source codes;
* Add enough unit tests to cover the possible use cases;
* Provide more comments per each function for readability, especially contracts that are verified in public;
* Provide more transparency on privileged activities once the protocol is live.

# Overview

## Project Summary

|  |  |
| --- | --- |
| Project Name | Swan Token |
| Platform | Solana |
| Codebase | https://github.com/ult-processor/SWAN\_AUDIT/tree/main/token |
| Commit | 7ce2d20bf8e3ac23882173aa4fbe68cd514821ac |

## Audit Summary

|  |  |
| --- | --- |
| Delivery Date | May 22, 2022 |
| Audit Methodology | Static Analysis, Manual Review |
| Key Components |  |

## Vulnerability Summary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vulnerability Level | Total | Pending | Acknowledged | Partially Resolved | Resolved |
| Critical | 0 | 0 | 0 | 0 | 0 |
| Major | 5 |  |  |  |  |
| Medium | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vulnerability Level | Total | Pending | Acknowledged | Partially Resolved | Resolved |
| Minor | 3 |  |  |  |  |
| Informational | 0 | 0 | 0 | 0 | 0 |
| Discussion | 0 | 0 | 0 | 0 | 0 |

## Audit Scope

|  |  |
| --- | --- |
| Cluster | Testnet |
| Mint Address | H27UjJhZvJ4obUChdgg8fbUzR2hCPB5JMWRVqvuEpjLk |

# FIndings

## Project Description

A token program which is forked from Solana Program Library, usable for fungible and non-fungible tokens.

This program provides an interface and implementation that third parties can utilize to create and use SWAN tokens.

## Project Structure



* cli: A basic command-line for creating and using SPL Tokens.
* js: Token javascript API.
* program: SPL token program core engine.
* program-2022: Latest SPL token program.
* rust: SPL token Rust client.
* ts: A Typescript library for interacting with the SPL Token program.

## Automated Test Using Cargo

* cargo test

It allows to run the host-based tests.

All passed.

There is no compiling errors or issues.

* cargo test-bpf

It allows BPF program tests.

All passed.

There is no compiling errors or issues.

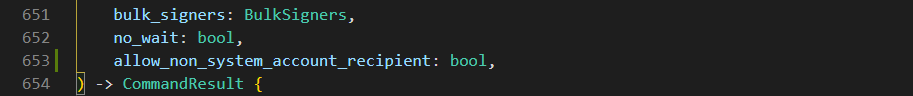
## Manual Test Comparing to SPL

* cli

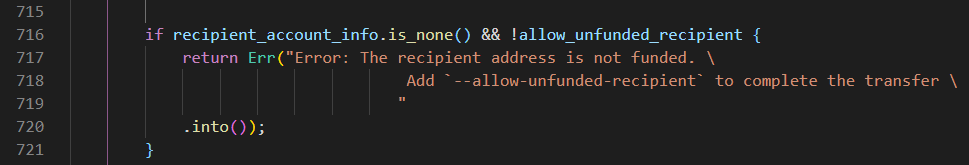
1. Missing “allow\_non\_system\_account\_recipient” parameter in row 653 of main.rs (Major Issue)



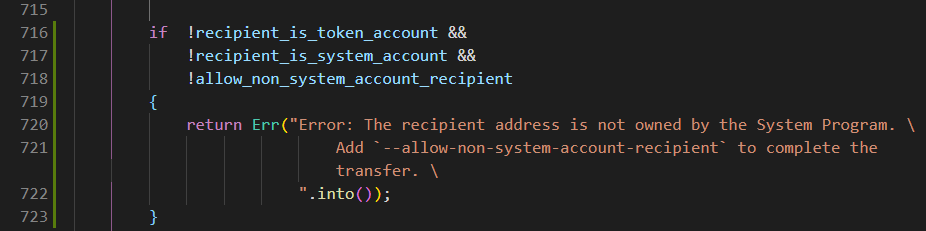
Should be added



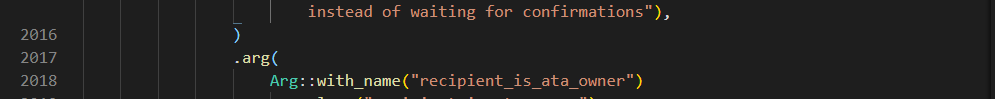
1. Missing “recipient\_is\_system\_account” assertion in row 715 of main.rs (Major Issue)



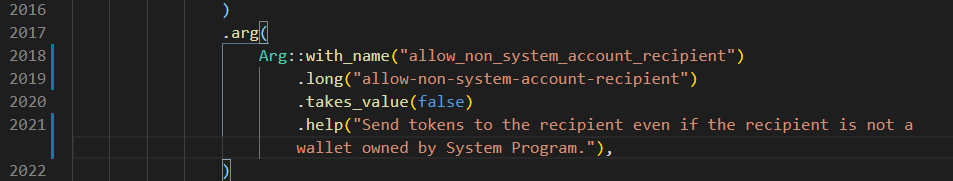
Should assert about the case of



1. Missing “allow\_non\_system\_account\_recepient” assertion in row 2017 of main.rs (Major Issue)



Should assert about the case of



1. Missing tests module in main.rs (Major Issue)

Should be added



* js

No issue founded.

* program

No issue founded.

* program-2022

1. Missing immutable\_owner module in src/extension (Minor Issue)

It is the module for Indicating that the Account owner authority cannot be changed

1. Missing non\_transferable module in src/extension (Minor Issue)

It is the module for Indicating that the tokens from this mint can't be transferred

1. Not implemented immutable\_owner and non\_transferable module in src/instruction.rs (Minor Issue)

* rust

No issue founded

* ts

No issue founded

* others

1. Missing program-2022-test module. (Major Issue)

Should be done SPL-Token 2022 Integration Tests.

**Recommendation:** Refer to latest spl token program tests.

## Others

* Hardcoded Addresses

Not found

* Honeypot Leak

Not found

# conclusion

## Overall Design

* Good project architecture and implemented spl token program standard requirements.

## Problems

* Missing some assertions and not implemented extensible option modules.

It could cause some unexpected results or inconvenience implementation in the future.

**Recommendation:** Refer to the latest spl token program and implement missing features.

# disclaimer

Note that this audit does not give any warranties on finding all possible security issues of given token program, i.e., the evaluation result does not guarantee the nonexistence of any further findings of security issues. As one audit-based assessment cannot be considered comprehensive, we always recommend proceeding with several independent audits and public bugs to ensure the security of this token program.

This token program is forked from spl token program so it could be involved unexpected security issues.

In this audit, we are concentrated on updated functionality security and common public bugs.