# MakerDAO: sUSDS Token

## Security Review

Cantina Managed review by:
**M4rio.eth**, Security Researcher
**Jonatas Martins**, Associate Security Researcher

September 26, 2024

# Contents

# 1 Introduction

## 1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at cantina.xyz

## 1.2 Disclaimer

Cantina Managed provides a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While Cantina Managed endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that the Cantina Managed security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

## 1.3 Risk assessment

| Severity | Description |
|---|---|
| **Critical** | *Must* fix as soon as possible (if already deployed). |
| **High** | Leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users. |
| **Medium** | Global losses <10% or losses to only a subset of users, but still unacceptable. |
| **Low** | Losses will be annoying but bearable. Applies to things like griefing attacks that can be easily repaired or even gas inefficiencies. |
| **Gas Optimization** | Suggestions around gas saving practices. |
| **Informational** | Suggestions around best practices or readability. |

### 1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

## 2 Security Review Summary

The Maker Protocol, also known as the Multi-Collateral Dai (MCD) system, allows users to generate Dai (a decentralized, unbiased, collateral-backed cryptocurrency soft-pegged to the US Dollar) by leveraging collateral assets approved by the Maker Governance, which is the community organized and operated process of managing the various aspects of the Maker Protocol.

From Sep 16th to Sep 17th the Cantina team reviewed the following source codes for MakerDAO holistically based on the corresponding commit hashes:

| Name | File Commit Hash |
|---|---|
| sUSDS Proxy | ERC1967Proxy.sol @ dbb6104c |
| sUSDS Implementation | SUsds.sol @ e1d160ab |
| sUSDS Deployment Lib | SUsdsDeploy.sol @ e1d160ab |

The L2 sUSDS implementation is of a standard UUPS upgradeable token and does not have staking logic. It is based on the USDS code.

Manual review will be done on the L2 token, and in some cases will replace the sanity checks that are usually done in init libs. For example, the following manual validations will be required from the team crafting the bridge initialization spell:

- Verify that calling `version()` on the proxy returns `1`.
- Verify that calling getImplementation on the proxy returns the address of the implementation contract.

The sUSDS token will be controlled by the L2 governance relay. Immediately after deployment, the L2 deployer grants admin control over the token to the L2 governance relay and revokes its own control.

Furthermore, its initialization flows were checked against `op-token-bridge` commit a01b8725 (it will be initialized at the same time as the bridge initialization). Note that initializing a token involves registering the token on L1 and L2 and authorising the L2 side of the bridge to mint tokens.

As a consequence of the aforementioned implementations, the token:

- Does not implement any particular getter that would, for example, link the token contract to the L2 side of bridge.
- Does not grant automatic burn approval to the L2 side of the bridge.

Note that these are some notable differences with the standard OP Stack L2 token implementation.

No issues were identified during the review.