PUBLIC

Code Assessment

of the DSS Blow2 Smart Contracts

February 21, 2025

Produced for

by

Contents

1 Executive Summary

Dear all,

Thank you for trusting us to help Sky with this security audit. Our executive summary provides an overview of subjects covered in our audit of the latest reviewed contracts of DSS Blow2 according to Scope to support you in forming an opinion on their security risks.

Sky implements DssBlow2, a contract to facilitate the returning of Dai and USDS to the Sky Protocol as system surplus.

The most critical subject covered in our audit is functional correctness.

The general subjects covered are event handling, documentation, and gas efficiency.

Security regarding all the aforementioned subjects is high.

In summary, we find that the codebase provides a high level of security.

It is important to note that security audits are time-boxed and cannot uncover all vulnerabilities. They complement but don't replace other vital measures to secure a project.

The following sections will give an overview of the system, our methodology, the issues uncovered, and how they have been addressed. We are happy to receive questions and feedback to improve our service.

Sincerely yours,

ChainSecurity

1.1 Overview of the Findings

Below we provide a brief numerical overview of the findings and how they have been addressed.

Critical-Severity Findings	0
High-Severity Findings	0
Medium-Severity Findings	0
Low-Severity Findings	0

2 Assessment Overview

In this section, we briefly describe the overall structure and scope of the engagement, including the code commit which is referenced throughout this report.

2.1 Scope

The assessment was performed on the source code files inside the DSS Blow2 repository based on the documentation files. The table below indicates the code versions relevant to this report and when they were received.

V	Date	Commit Hash	Note
1	10 Feb 2025	d4e65edcc60aee324eab1b3e6509f91a0f831325	Initial Version

For the solidity smart contracts, the compiler version `0.8.24` was chosen.

The following contracts were in scope:

```
src/
    DssBlow2.sol
deployment/
    DssBlow2Deploy.sol
    DssBlow2Instance.sol
```

2.1.1 Excluded from scope

Generally, all files not mentioned above are out of scope.

2.2 System Overview

This system overview describes the initially received version (Version 1) of the contracts as defined in the Assessment Overview.

Furthermore, in the findings section, we have added a version icon to each of the findings to increase the readability of the report.

Sky offers DssBlow2, a contract to facilitate the returning of Dai and USDS to the Sky Protocol as system surplus.

2.2.1 DssBlow2

The permissionless public function `blow` is the main entrypoint of this contract. If there is any available DAI or USDS balance in this contract, it calls function `join` on the respective token's JOIN adapter where the tokens are burned and the VOW's (System Surplus Buffer) DAI balance in the VAT is increased.

To return DAI or USDS tokens to the Sky Protocol, users can simply do ERC-20 transfers of both tokens to this contract. Later, function `blow` can be called by anyone to add both the outstanding Dai and USDS balances to the VOW at the same time.

2.2.2 Deployment

Library DssBlow2Deploy provides a function `deploy` that deploys an instance of DssBlow2 with the address of VOW, DAI and USDS JOINs. In the constructor, contract DssBlow2 will be configured with VOW, DAI / USDS and the respective JOINs. In addition, allowances of `type(uint256).max` (infinite allowances) DAI and USDS will be approved to the respective JOINs, hence the JOINs can burn the tokens on behalf of the contract.

2.2.3 Roles & Trust Model

DssBlow2 is permissionless without any privileged roles and immutable once deployed.

It is assumed the deployer will deploy DssBlow2 with correct addresses and the addresses should be checked before the deployed contract is used.

3 Limitations and use of report

Security assessments cannot uncover all existing vulnerabilities; even an assessment in which no vulnerabilities are found is not a guarantee of a secure system. However, code assessments enable the discovery of vulnerabilities that were overlooked during development and areas where additional security measures are necessary. In most cases, applications are either fully protected against a certain type of attack, or they are completely unprotected against it. Some of the issues may affect the entire application, while some lack protection only in certain areas. This is why we carry out a source code assessment aimed at determining all locations that need to be fixed. Within the customer-determined time frame, ChainSecurity has performed an assessment in order to discover as many vulnerabilities as possible.

The focus of our assessment was limited to the code parts defined in the engagement letter. We assessed whether the project follows the provided specifications. These assessments are based on the provided threat model and trust assumptions. We draw attention to the fact that due to inherent limitations in any software development process and software product, an inherent risk exists that even major failures or malfunctions can remain undetected. Further uncertainties exist in any software product or application used during the development, which itself cannot be free from any error or failures. These preconditions can have an impact on the system's code and/or functions and/or operation. We did not assess the underlying third-party infrastructure which adds further inherent risks as we rely on the correct execution of the included third-party technology stack itself. Report readers should also take into account that over the life cycle of any software, changes to the product itself or to the environment in which it is operated can have an impact leading to operational behaviors other than those initially determined in the business specification.

4 Terminology

For the purpose of this assessment, we adopt the following terminology. To classify the severity of our findings, we determine the likelihood and impact (according to the CVSS risk rating methodology).

- *Likelihood* represents the likelihood of a finding to be triggered or exploited in practice
- *Impact* specifies the technical and business-related consequences of a finding
- *Severity* is derived based on the likelihood and the impact

We categorize the findings into four distinct categories, depending on their severity. These severities are derived from the likelihood and the impact using the following table, following a standard risk assessment procedure.

Likelihood	Impact		
	High	Medium	Low
High	Critical	High	Medium
Medium	High	Medium	Low
Low	Medium	Low	Low

As seen in the table above, findings that have both a high likelihood and a high impact are classified as critical. Intuitively, such findings are likely to be triggered and cause significant disruption. Overall, the severity correlates with the associated risk. However, every finding's risk should always be closely checked, regardless of severity.

5 Open Findings

In this section, we describe our findings. The findings are split into these different categories:

Below we provide a numerical overview of the identified findings, split up by their severity.

Critical-Severity Findings	0
High-Severity Findings	0
Medium-Severity Findings	0
Low-Severity Findings	0

6 Notes

We leverage this section to highlight further findings that are not necessarily issues. The mentioned topics serve to clarify or support the report, but do not require an immediate modification inside the project. Instead, they should raise awareness in order to improve the overall understanding.

6.1 Difference Between DssBlow and DssBlow2

Note Version 1

Users should be aware of the key differences between DssBlow2 and DssBlow:

1. DssBlow only supports DAI tokens while DssBlow2 supports both DAI and USDS.
2. DssBlow provides an additional interface, `blow(uint256 wad)`, which combines a DAI transfer from the caller with the `join()` operation in a single call and therefore requires prior approval for the caller's DAI tokens. This function is not available in DssBlow2.

6.2 Funds in DssBlow2 May Not Be Accounted in a Shutdown

Note Version 1

In the process of a global shutdown (END), function `thaw` will fix the DAI supply (`debt`) which checks the system surplus is cleared `vat.dai(address(vow)) == 0`. If surplus DAI/USDS remains in DssBlow2 and is not transferred to VOW before thaw is called, this amount will not be accounted for in reducing the debt, potentially leading to users being able to cash out less collateral.

Since DssBlow2 is not a core system contract but simply an external helper for returning funds to the system, these funds are not automatically considered by the system. Any surplus in DssBlow2 should be pushed to the VOW in time before shutdown and no further tokens should be transferred to DssBlow2 afterward.

6.3 Other Tokens Sent to DssBlow2 Will Be Locked

Note Version 1

DssBlow2 contract can only handle DAI and USDS tokens. In case any other tokens are transferred to this contract, they will be locked forever.