Disa Token Security Assessment

Findings and Recommendations Report Presented to:

Disatok

June 23, 2022

Version: 3.0

Presented by:

Kudelski Security, Inc.
5090 North 40th Street, Suite 450
Phoenix, Arizona 85018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Overview

Disatok engaged Kudelski Security to perform a security assessment of the Disa Token smart contracts.

The assessment was conducted remotely by the Kudelski Security Team. Testing took place on April 19 - May 02, 2022, and focused on the following objectives:

- Provide the customer with an assessment of their overall security posture and any risks that were discovered within the environment during the engagement.
- To provide a professional opinion on the maturity, adequacy, and efficiency of the security measures that are in place.
- To identify potential issues and include improvement recommendations based on the result of our tests.

This report summarizes the engagement, tests performed, and findings. It also contains detailed descriptions of the discovered vulnerabilities, steps the Kudelski Security Teams took to identify and validate each issue, as well as any applicable recommendations for remediation.

Key Findings

The following are the major themes and issues identified during the testing period. These, along with other items, within the findings section, should be prioritized for remediation to reduce the risk they pose:

- KS-DISATOK-01 – DISATOK.sol - Indirect transfers don't take fees into account
- KS-DISATOK-02 – TokenFarm.sol - Order of interest calculation leads to 0 rewards
- KS-DISATOK-03 – TokenFarm.sol - Rewards double spending
- KS-DISATOK-04 – TokenFarm.sol - Test function is available to the public

During the test, the following positive observations were noted regarding the scope of the engagement:

- The team was very supportive and open to discussing the design choices made

Based on formal verification we conclude that the reviewed code implements the documented functionality.

Scope and Rules of Engagement

Kudelski performed a Disa Token Security Assessment. The following table documents the targets in scope for the engagement. No additional systems or resources were in scope for this assessment.

The source code was supplied through a public repository at https://github.com/Tuleva-AG/disatok with the commit hash 8e4a90ab9657f5a1a8c0a742cf1728df74079c39. A re-review was performed on May 30, 2022, with the commit hash 6490377493a8a7077a3b0c81d7e89c6ca2dcb502.

Files included in the code review
disatok/ ├── contracts/ │ └── Ownable.sol ├── extensions/ │ └── IERC20Metadata.sol ├── utils/ │ └── Context.sol ├── DISATOK.sol ├── IERC20.sol └── TokenFarm.sol

Table 1: Scope

TECHNICAL ANALYSIS & FINDINGS

During the Disa Token Security Assessment, we discovered:

- 4 findings with HIGH severity rating.
- 6 findings with MEDIUM severity rating.
- 3 findings with LOW severity rating.
- 3 findings with INFORMATIONAL severity rating.

The following chart displays the findings by severity.

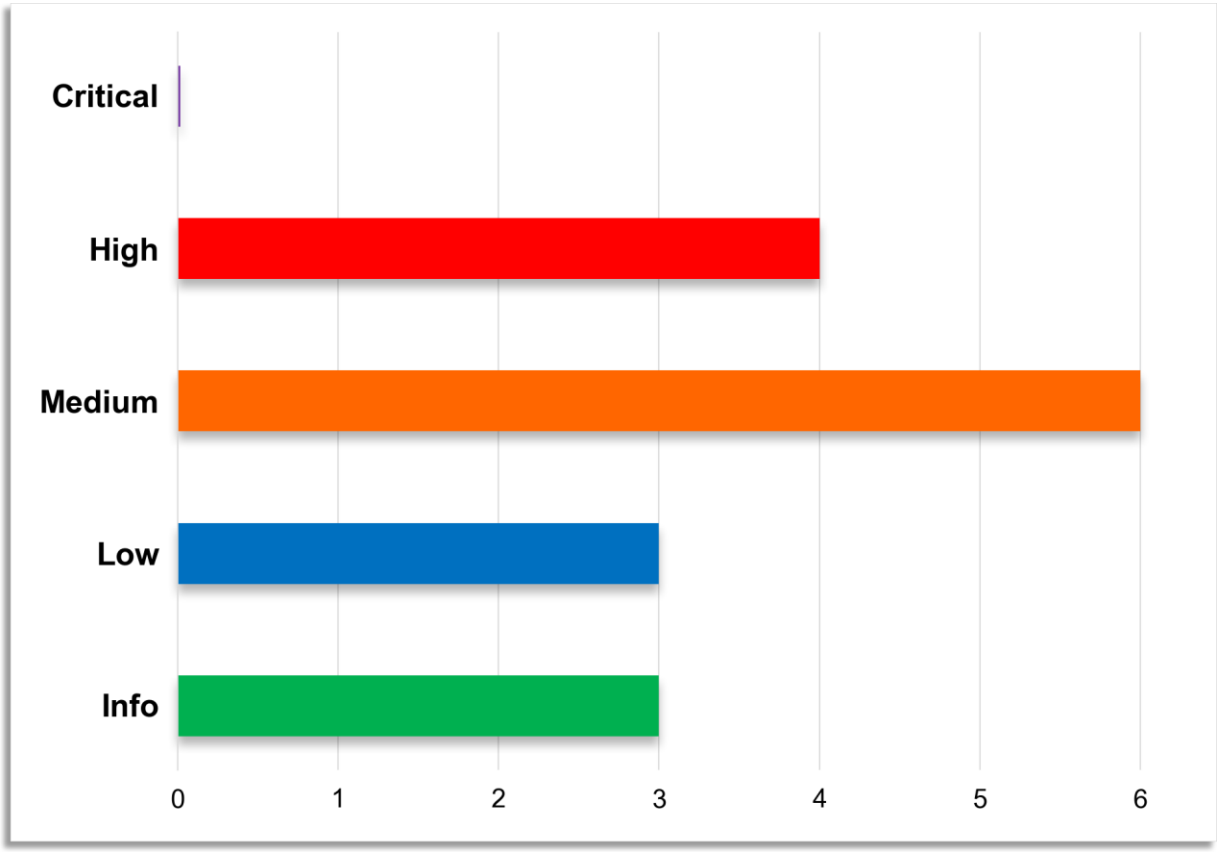

Figure 1: Findings by Severity

Findings

The *Findings* section provides detailed information on each of the findings, including methods of discovery, explanation of severity determination, recommendations, and applicable references.

After further discussion with the Disatok team, the following finding was determined to be not an issue and has been removed:

- KS-DISATOK-14 - DISATOK.sol - Tokens are not automatically transferred

The following table provides an overview of the findings.

#	Severity	Description	Status
KS-DISATOK-01	High	DISATOK.sol - Indirect transfers don't take fees into account	Remediated
KS-DISATOK-02	High	TokenFarm.sol - Order of interest calculation leads to 0 rewards	Remediated
KS-DISATOK-03	High	TokenFarm.sol - Rewards double spending	Remediated
KS-DISATOK-04	High	TokenFarm.sol - Test function is available to the public	Remediated
KS-DISATOK-05	Medium	DISATOK.sol - Inconsistent data in events emitted during token transfer	Remediated
KS-DISATOK-06	Medium	DISATOK.sol - Ownership can be renounced	Remediated
KS-DISATOK-07	Medium	DISATOK.sol - Sales account fee exclusion is not updated	Remediated
KS-DISATOK-08	Medium	TokenFarm.sol - Batch reward payouts can be expensive	Remediated
KS-DISATOK-09	Medium	TokenFarm.sol - Contract does not check for sufficient balance	Remediated
KS-DISATOK-10	Medium	TokenFarm.sol - User's balance is not updated during token issuance	Remediated
KS-DISATOK-11	Low	DISATOK.sol - Gas expendature	Remediated
KS-DISATOK-12	Low	DISATOK.sol - Tokens sent to the contract can be locked	Remediated
KS-DISATOK-13	Low	DISATOK.sol - Updating ownership does not update fee exclusion	Remediated
KS-DISATOK-15	Informational	DISATOK.sol - Use of hard-coded values	Remediated
KS-DISATOK-16	Informational	Missing event emissions	Remediated
KS-DISATOK-17	Informational	Unused variables	Remediated

Table 2: Findings Overview

Technical Analysis

The source code has been manually validated to the extent that the state of the repository allowed. The validation includes confirming that the code correctly implements the intended functionality. Based on formal verification we conclude that the code implements the documented functionality to the extent of the reviewed code

Technical Findings

General Observations

Disatok offers an ERC20 Token named DISATOK, along with a TokenFarm contract which allows owners of DISATOK to stake tokens and receive interest depending on the length of time their tokens are staked. The project seems to be in its early stage. No tests were provided while the source code contained quite a bit of commented functionality, which was ignored during the audit. Being its early stage the project also misses some guarantees, which are described in the findings below. The development team was quick to address, communicate, and comment on our observations during the auditing process.

METHODOLOGY

Kudelski Security uses the following high-level methodology when approaching engagements. They are broken up into the following phases.

Figure 2: Methodology Flow

Kickoff

The project is kicked off as the sales process has concluded. We typically set up a kickoff meeting where project stakeholders are gathered to discuss the project as well as the responsibilities of participants. During this meeting we verify the scope of the engagement and discuss the project activities. It's an opportunity for both sides to ask questions and get to know each other. By the end of the kickoff there is an understanding of the following:

- Designated points of contact
- Communication methods and frequency
- Shared documentation
- Code and/or any other artifacts necessary for project success
- Follow-up meeting schedule, such as a technical walkthrough
- Understanding of timeline and duration

Ramp-up

Ramp-up consists of the activities necessary to gain proficiency on the particular project. This can include the steps needed for familiarity with the codebase or technological innovation utilized. This may include, but is not limited to:

- Reviewing previous work in the area including academic papers
- Reviewing programming language constructs for specific languages
- Researching common flaws and recent technological advancements

Review

The review phase is where most of the work on the engagement is completed. This is the phase where we analyze the project for flaws and issues that impact the security posture. Depending on the project this may include an analysis of the architecture, a review of the code, and a specification matching to match the architecture to the implemented code.

In this code audit, we performed the following tasks:

1. Security analysis and architecture review of the original protocol
2. Review of the code written for the project

3. Compliance of the code with the provided technical documentation

The review for this project was performed using manual methods and utilizing the experience of the reviewer. No dynamic testing was performed, only the use of custom-built scripts and tools were used to assist the reviewer during the testing. We discuss our methodology in more detail in the following sections.

Code Safety

We analyzed the provided code, checking for issues related to the following categories:

- General code safety and susceptibility to known issues
- Poor coding practices and unsafe behavior
- Leakage of secrets or other sensitive data through memory mismanagement
- Susceptibility to misuse and system errors
- Error management and logging

This list is general list and not comprehensive, meant only to give an understanding of the issues we are looking for.

Technical Specification Matching

We analyzed the provided documentation and checked that the code matches the specification. We checked for things such as:

- Proper implementation of the documented protocol phases
- Proper error handling
- Adherence to the protocol logical description

Reporting

Kudelski Security delivers a preliminary report in PDF format that contains an executive summary, technical details, and observations about the project.

The executive summary contains an overview of the engagement including the number of findings as well as a statement about our general risk assessment of the project. We may conclude that the overall risk is low but depending on what was assessed we may conclude that more scrutiny of the project is needed.

We not only report security issues identified but also informational findings for improvement categorized into several buckets:

- Critical
- High
- Medium
- Low
- Informational

The technical details are aimed more at developers, describing the issues, the severity ranking and recommendations for mitigation.

As we perform the audit, we may identify issues that aren't security related, but are general best practices and steps, that can be taken to lower the attack surface of the project. We will call those out as we encounter them and as time permits.

As an optional step, we can agree on the creation of a public report that can be shared and distributed with a larger audience.

Verify

After the preliminary findings have been delivered, this could be in the form of the approved communication channel or delivery of the draft report, we will verify any fixes withing a window of time specified in the project. After the fixes have been verified, we will change the status of the finding in the report from open to remediated.

The output of this phase will be a final report with any mitigated findings noted.

Additional Note

It is important to note that, although we did our best in our analysis, no code audit or assessment is a guarantee of the absence of flaws. Our effort was constrained by resource and time limits along with the scope of the agreement.

While assessment the severity of the findings, we considered the impact, ease of exploitability, and the probability of attack. These is a solid baseline for severity determination.

The Classification of identified problems and vulnerabilities

There are four severity levels of an identified security vulnerability.

Critical – vulnerability that will lead to loss of protected assets

- This is a vulnerability that would lead to immediate loss of protected assets
- The complexity to exploit is low
- The probability of exploit is high

High - A vulnerability that can lead to loss of protected assets

- All discrepancies found where there is a security claim made in the documentation that cannot be found in the code
- All mismatches from the stated and actual functionality
- Unprotected key material
- Weak encryption of keys
- Badly generated key materials
- Tx signatures not verified
- Spending of funds through logic errors
- Calculation errors overflows and underflows

Medium - a vulnerability that hampers the uptime of the system or can lead to other problems

- Insecure calls to third party libraries
- Use of untested or nonstandard or non-peer-revied crypto functions
- Program crashes leaves core dumps or write sensitive data to log files

Low - Problems that have a security impact but does not directly impact the protected assets

- Overly complex functions
- Unchecked return values from 3rd party libraries that could alter the execution flow

Informational

- General recommendations