ChainLight

DELABS Token Security Audit

：DELABS Token

May 23, 2025

Revision 2.0

ChainLight@Theori

Table of Contents

Executive Summary

(v1.0)

Starting July 15, 2024, ChainLight of Theori audited DELABS Token smart contracts for a day. In the audit, we primarily considered the issues/impacts listed below.

- Theft of funds
- Permanent freeze of funds
- Insufficient access control
- Single point of failure
- Incorrect usage/deployment of well-known libraries

As a result, we identified the issues listed below.

- Total: 3
- Informational: 3 (Potential SPOFs, A weakness that might lead to permanent freeze of funds on specific conditions)

(v2.0)

Beginning on May 21, 2025, ChainLight of Theori conducted a day security audit of the updated version (OFT Integration) of DELABS Token.

As a result, we identified the issues listed below.

- Total: 1
- Informational: 1

Audit Overview

Scope

Name	DELABS Token Security Audit
Target / Version	• `delabs-token-share-master.zip` : sha256sum `1b5711824fbec80d6d346dc40e5c3836c62fa8bfaabf6cb1f0ba3d89420f1d4f` • (v2.0) `DELABS.sol` : sha256sum `637c96f552c524daf0ea1f103cadc4ade3753be4131b8eaafb6f992a5f58089f`
Application Type	Smart contracts
Lang. / Platforms	Smart contracts [Solidity]

Code Revision

v2.0: OFT Integration

Severity Categories

Severity	Description
Critical	The attack cost is low (not requiring much time or effort to succeed in the actual attack), and the vulnerability causes a high-impact issue. (e.g., Effect on service availability, Attacker taking financial gain)
High	An attacker can succeed in an attack which clearly causes problems in the service's operation. Even when the attack cost is high, the severity of the issue is considered "high" if the impact of the attack is remarkably high.
Medium	An attacker may perform an unintended action in the service, and the action may impact service operation. However, there are some restrictions for the actual attack to succeed.
Low	An attacker can perform an unintended action in the service, but the action does not cause significant impact or the success rate of the attack is remarkably low.
Informational	Any informational findings that do not directly impact the user or the protocol.
Note	Neutral information about the target that is not directly related to the project's safety and security.

Status Categories

Status	Description
Reported	ChainLight reported the issue to the client.
WIP	The client is working on the patch.
Patched	The client fully resolved the issue by patching the root cause.
Mitigated	The client resolved the issue by reducing the risk to an acceptable level by introducing mitigations.
Acknowledged	The client acknowledged the potential risk, but they will resolve it later.
Won't Fix	The client acknowledged the potential risk, but they decided to accept the risk.

Finding Breakdown by Severity

Category	Count	Findings
Critical	0	• N/A
High	0	• N/A
Medium	0	• N/A
Low	0	• N/A
Informational	4	• `DELABS2407-001` • `DELABS2407-002` • `DELABS2407-003` • `DELABS2407-004`
Note	0	• N/A

Findings

Summary

#	ID	Title	Severity	Status
1	DELABS2407-001	Add `_disableInitializers()` call to `DELABS.constructor()`	Informational	Patched
2	DELABS2407-002	The initial owner of the `DELABS` contract should be a timelock	Informational	Patched
3	DELABS2407-003	Recommendations on timelock deployment and configuration	Informational	Mitigated
4	DELABS2407-004	Inclusion of Testnet Chain ID in `DELABS.initialize()` Affects Code Clarity	Informational	Acknowledged

#1 `DELABS2407-001` Add `_disableInitializers()` call to `DELABS.constructor()`

ID	Summary	Severity
`DELABS2407-001`	Not calling `_disableInitializers()` in the `constructor()` may lead to permanent loss of funds under specific conditions not currently met.	Informational

Description

Currently, the `DELABS` contract does not have a `constructor()` that calls `_disableInitializers()`. While there is no immediate risk, future code updates could allow `initialize()` calls to lead to a `delegatecall()` to an arbitrary address. If this occurs on a network where EIP-6780 is not implemented, an attacker could use `selfdestruct()` to delete the implementation contract. After the attack, given the use of `UUPSUpgradeable`, recovering the contract through upgrades would be impossible.

Impact

Informational

If the two preconditions mentioned in the description are met, it would lead to a permanent loss of funds. However, these conditions are not currently met.

Recommendation

Invoke `_disableInitializers()` within the `DELABS.constructor()` to prevent the initializer function from being called on the implementation contract.

Remediation

Patched

The issue has been patched as recommended.

#2 DELABS2407-002 The initial owner of the DELABS contract should be a timelock

ID	Summary	Severity
DELABS2407-002	The MyTimelockController contract should be the owner of the DELABS contract. However, the current deployment procedure would assign ownership to an externally owned account (EOA), which contradicts the intended design.	Informational

Description

The DELABS contract sets the initial owner to the msg.sender, which is the externally owned account (EOA) that deployed the contract. Although the EOA can transfer ownership to MyTimelockController, this action cannot be performed immediately if the timelock is appropriately configured, as it would require calling acceptOwnership() from the timelock, which is subject to delays.

Impact

Informational

The DELABS contract's owner, who has complete control of the contract due to the upgrade feature, is set to an EOA for some time, needlessly introducing a temporary single point of failure (SPOF).

Recommendation

Modify the initialize() to accept the address of MyTimelockController as a parameter and immediately assign it as the owner. This ensures that the MyTimelockController is set as the owner at the time of deployment, avoiding the need for subsequent ownership transfers.

Remediation

Patched

The issue has been patched as recommended.

#3 `DELABS2407-003` Recommendations on timelock deployment and configuration

ID	Summary	Severity
`DELABS2407-003`	`optional admin` privileges must be renounced promptly after the initial setup. This procedure is not included in the deployment script or the documentation.	Informational

Description

After deploying the `MyTimelockController`, renouncing the `optional admin` privileges is crucial as soon as the initial configuration is completed. This measure ensures that all subsequent modifications to the timelock roles are executed via a timelock proposal, thereby enhancing security. The current deployment script does not include this procedure, and no comprehensive plan is outlined in the provided documentation.

Impact

Informational

Failure to renounce the `optional admin` privileges may lead to a Single Point of Failure (SPOF) scenario where an attacker compromises the `optional admin` EOA, grants required roles to themselves, revokes existing roles, and then queues and executes transactions without intervention.

Recommendation

`optional admin` privileges must be renounced promptly after the initial setup. This can be done by updating the deployment script or the operational procedure.

Remediation

Mitigated

Delabs team provided the deployment plan regarding the timelock, and we could verify that it is appropriate.

#4 `DELABS2407-004` Inclusion of Testnet Chain ID in `DELABS.initialize()` Affects Code Clarity

ID	Summary	Severity
DELABS2407-004	The `DELABS` contract's `initialize` function includes a hardcoded Sepolia testnet chain ID (`11155111`) alongside the Ethereum mainnet ID (`1`) in its initial token minting condition. While tokens minted on testnet would have no bearing on mainnet tokens, the presence of two chain IDs in the code reduces clarity regarding the intended total supply.	Informational

Description

The `DELABS` contract's `initialize` function uses a conditional statement to mint the `TOTAL_SUPPLY`:

```
    if (block.chainid == 1 || block.chainid == 11155111) {
        _mint(initialHolder, TOTAL_SUPPLY);
    }
```

This logic permits initial minting if the `block.chainid` is `1` (Ethereum mainnet) or `11155111` (Sepolia testnet). Client clarification confirms the Sepolia ID is for testing and is intended to be removed for the final mainnet deployment.

While tokens minted on a distinct testnet (like Sepolia, via the `11155111` condition) would be isolated and would not affect the value or supply of tokens on the mainnet, the current dual condition in the code can lead to misinterpretation. Specifically, without careful analysis of the `block.chainid`, a reader might incorrectly infer that the `TOTAL_SUPPLY` is intended to be minted on multiple networks.

Impact

Informational

This affects code clarity and maturity **and** may create confusion for developers and auditors.

Recommendation

For the mainnet deployment, the conditional statement in the `initialize` function should be updated to `if (block.chainid == 1)`, thereby removing the Sepolia testnet chain ID.

Remediation

Acknowledged

Removal of the testnet chain ID was already planned and will be executed before the mainnet deployment.

Revision History

Version	Date	Description
1.0	July 26, 2024	Initial version
2.0	May 23, 2025	OFT Integration

Theori, Inc. (“We”) is acting solely for the client and is not responsible to any other party. Deliverables are valid for and should be used solely in connection with the purpose for which they were prepared as set out in our engagement agreement. You should not refer to or use our name or advice for any other purpose. The information (where appropriate) has not been verified. No representation or warranty is given as to accuracy, completeness or correctness of information in the Deliverables, any document, or any other information made available. Deliverables are for the internal use of the client and may not be used or relied upon by any person or entity other than the client. Deliverables are confidential and are not to be provided, without our authorization (preferably written), to entities or representatives of entities (including employees) that are not the client, including affiliates or representatives of affiliates of the client.