ChainLight

HyperToken Security Audit

: Hyperlane - HyperToken

Apr 18, 2025

Revision 1.0

ChainLight@Theori

Table of Contents

Executive Summary

From **31 March 2025**, ChainLight (Theori) performed a security audit of the **HyperToken** smart-contract. The engagement focused on uncovering critical vulnerabilities and assessing their potential impact.

Thanks to the contract's compact codebase and comprehensive test coverage—including fuzzing—only a single, **informational-severity** issue was discovered.

Summary of Findings

- **Informational:** 1

Audit Overview

Scope

Name	HyperToken Security Audit
Target / Version	• Git Repository (`hyperlane-xyz/hyperlane-monorepo-private`): PR `#23`, commit `db9edf0379459fdc566e5debe6d4688e2e5f541e`
Application Type	Smart contracts
Lang. / Platforms	Smart contracts [Solidity]

Code Revision

N/A

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	1	• `HYPERTOKEN-001`
Note	0	• N/A

Findings

Summary

#	ID	Title	Severity	Status
1	HYPERTOKEN-001	Missing `__ERC20Permit_init` Call in `HyperToken.initialize()`	Informational	Patched

#1 HYPERTOKEN-001 Missing __ERC20Permit_init Call in HyperToken.initialize()

ID	Summary	Severity
HYPERTOKEN-001	HyperToken.initialize() omits the call to __ERC20Permit_init(_name). Consequently, the EIP-712 domain separator is built with empty name and version fields. Although permit() still works, signatures must be generated against this partially empty domain.	Informational

Description

HyperToken extends HypERC20, which in turn inherits from ERC20PermitUpgradeable. The parent initializer __ERC20Permit_init(string) stores the token's name and the constant version "1" in EIP-712 storage. Because the call is missing, these fields remain unset and _domainSeparatorV4() hashes empty strings for both. Off-chain tools therefore have to sign with an empty name and version; otherwise, signature verification in permit() fails. While chainId and verifyingContract are still included, deviating from the expected domain format is considered poor practice.

Impact

Informational

Signers receive less contextual information in their wallet, increasing susceptibility to phishing or social-engineering attacks.

Recommendation

Add the missing initializer call. (__ERC20Permit_init(_name);)

Remediation

Patched

It has been patched as recommended.

Revision History

Version	Date	Description
1.0	Apr 18, 2025	Initial version

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.