Protocol Audit Report

Version 1.0

github.com/Zyrrow

October 26, 2024

Protocol Audit Report

Zyrrow

October 26 , 2024

Prepared by: [Zyrrow] Lead Auditors:

- Zyrrow

Table of Contents

Protocol Summary

A smart contract application for storing a password. Users should be able to store a password and then retrieve it later. Others should not be able to access the password.

Disclaimer

Zyrrow makes all effort to find as many vulnerabilities in the code in the given time period, but holds no responsibilities for the findings provided in this document. A security audit by me is not an endorsement of the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the Solidity implementation of the contracts.

Risk Classification

		Impact		
		High	Medium	Low
	High	H	H/M	M
Likelihood	Medium	H/M	M	M/L
	Low	M	M/L	L

We use the CodeHawks severity matrix to determine severity. See the documentation for more details.

Audit Details

Commit Hash: 7d55682ddc4301a7b13ae9413095feffd9924566

Scope

./src/

- PasswordStore.sol

Roles

-Owner: The user who can set the password and read the password. -Outsides: No one else should be able to set or read the password.

Executive Summary

This report details security vulnerabilities identified in the PasswordStore contract. The discovered issues compromise the confidentiality of sensitive information and the integrity of specific contract functions. Security flaws include insufficient access control, improper handling of sensitive data visibility, and documentation errors, all of which impact the contract's security and reliability.

Issues found

Severity	Issue
High	Storing the password in plain text on the blockchain
High	Lack of access control on the `setPassword` function
Informational	Incorrect natspec documentation

Findings

Actually 2 vulnerabilities found and 1 informationnal issue.

High

[H-1] Storing the password in plain text on the blockchain Description: The password is directly stored on-chain in the s_password variable, making it visible to any user. Impact: Any user can read the password, severely compromising the protocol's security. Recommended Mitigation: Store only an encrypted version of the password on-chain and avoid using a getter function to reduce the risk of leakage. [H-2] No Access Control on the setPassword Function Description: The setPassword function is set as external, allowing any user to change the password, though it should only be accessible by the contract owner. Impact: Any user can set/change the stored password, undermining the contract's

intended functionality. Recommended Mitigation: Implement an access control check to restrict password setting to the contract owner only.

Medium

Low

Informational

[I-1] Incorrect Natspec Documentation Description: The getPassword function's natspec indicates a newPassword parameter that does not exist in the function's signature, resulting in an inaccurate natspec. Impact: The inaccurate documentation could cause confusion for users and developers. Recommended Mitigation: Remove the incorrect @param line from the function's natspec.

Gas