Protocol Audit Report

Author: David Date: December 26, 2023

Table of Contents

Protocol Summary

The protocol is designed for storing and retrieving passwords. The owner should be able to store passwords and then retrieve them later. Others should not be able to access the password.

Disclaimer

The team makes all efforts 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 the team 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: Sets and retrieves passwords Users: Can't set or retrieve password

Executive Summary

We spent 80 hours auditing each and every storage and external calls, we used foundry, slither and hardhard to test every potential points of entry. We found that the private keyword is not really private, and access control was not made.

Issues Found

Severity	Number of Issues
High	2
Medium	0
Low	0
Info	1
Total	3

Findings

High

H-1

Security Risks of Storing Passwords on Blockchain. Storing passwords on a blockchain exposes them to public visibility, compromising privacy even with the “private” keyword.

Description: Despite the use of privacy measures such as the “private” keyword, the practice exposes passwords to public visibility, raising concerns about the confidentiality of sensitive information.

Impact: Any one can access and break the function.

Proof of Concept:

Recommended Mitigation:

H-2

TITLE Root Cause + Impact

`PasswordStored::s_password` Function has no access control, any one can call and change password

Description: `PasswordStored::s_password` function is external and restrictions needs to be made to allow only owner to call this and set it

Impact: There are no access control of the contract function and can break the contract through this

Proof of Concept:

Recommended Mitigation:

Medium

Low

Informational

I-1

TITLE Root Cause + Impact

The `passwordStore::getPassword` natspec indicates a parameter that doesn't exist causing the natspec to be incorrect

Description:

Impact: The natspec is incorrect.

Proof of Concept:

Recommended Mitigation:

Gas