Prepared by:

- Author: Claudia Romila
- Title: Protocol Audit Report
- Date: 10 January 2025

Table of Contents

Protocol Summary

PasswordStore is a protocol dedicated to storage and retrieval of a user's passwords. The protocol is designed to be used by a single user, and is not designed to be used by multiple users. Only the owner should be able to set and access this password.

Disclaimer

Claudia 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 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

Audit Details

- Commit hash:

Scope

```
./src/PasswordStore-sol
```

Roles

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

Executive Summary

- I dedicated 1 hour to completing the analysis and reporting process, utilizing available resources effectively.

Issues found

Severity	Number of issues found
High	1
Medium	0
Low	0
Info	1
Total	2

Findings

High

[H-1] `PasswordStore::setPassword` has no access control, meaning no-owner can change the password.

Description: `PasswordStore::setPassword` function is set to external, however, the NatSpec of the function and overall of the smart contract is that This function should only allow the owner to set a new password.

▶ Code

```
function setPassword(string memory newPassword) external  {
@>  //@audit Missing access control
       s_password = newPassword;
       emit SetNewPassword();
}
```

Impact: Anyone can set/change the password of the contract, severely breaking the contract's intended functionality.

Proof of Concept: Add the following to the `PasswordStore.t.sol` test file.

Code

```
function setPassword(string memory newPassword) external  {
@>  //@audit Missing access control
        s_password = newPassword;
        emit SetNewPassword();
 }
```

Recommandation: Add access control conditional to the `setPassword` function.

```
if(msg.sender != s_owner) {
    revert PasswordStore__NotOwner();
}
```

Informational

[I-1] `PasswordStore::getPassword` natspec indicates a parameter that dasn't exists, causing the natspec to be incorrect.

Impact: The natspec is incorrect.

Recommandation: Remove the incorrect natspec line.

```diff
-    * @param newPassword The new password to set.
```